Improving Adversarial Robustness in 3D Point Cloud Classification via Self-Supervisions

Jiachen Sun [1] Yulong Cao [1] Christopher Choy [2] Zhiding Yu [2] Chaowei Xiao [2] Anima Anandkumar [2]
Z. Morley Mao [1]

Abstract

3D point cloud data is increasingly used in safety-critical applications such as autonomous driving. Thus, robustness of 3D deep learning models against adversarial attacks is a major consideration. In this paper, we systematically study the impact of various self-supervised learning proxy tasks on different architectures and threat models for 3D point clouds. Specifically, we study MLP-based (PointNet), convolution-based (DGCNN), and transformer-based (PCT) 3D architectures. Through comprehensive experiments, we demonstrate that appropriate self-supervisions can significantly enhance the robustness in 3D point cloud recognition, achieving considerable improvements compared to the standard adversarial training baseline. Our analysis reveals that *local* feature learning is desirable for adversarial robustness since it limits the adversarial propagation between the point-level input perturbations and the model's final output. It also explains the success of DGCNN and the jigsaw proxy task in achieving 3D robustness.

1. Introduction

Point cloud data is one of the most widely used representations in 3D computer vision. It is a versatile data format available from various sensors and computer-aided design (CAD) models. Given such advantages, many deep learning-based 3D perception systems have been proposed (Choy et al., 2019; Maturana & Scherer, 2015; Qi et al., 2017; Riegler et al., 2017; Wang & Posner, 2015; Wang et al., 2017) and achieved great success in safety-critical applications (*e.g.,* autonomous driving) (Shi et al., 2019; 2020; Yin et al., 2021). Although deep learning on point clouds has exhibited high performance, they are particularly vulnerable to adversarial attacks (Cao et al., 2019; Sun et al., 2020a; Xiang et al., 2019). Because of the wide applications in safety-critical fields, it is imperative to study the adversarial robustness of point cloud recognition models.

Self-supervised learning (SSL) has been incorporated into adversarial training (AT) in 2D image perception models lately. It has shown great potential to enhance adversarial robustness without requiring any additional data or labels (Chen et al., 2020; Hendrycks et al., 2019). Given such achievements, a natural question emerges: can we mimic the application of SSL to improve adversarial robustness in 3D point cloud recognition? Such a label-free strategy is strongly preferred due to the cost and difficulty of 3D point cloud data annotation (Qi et al., 2021).

Summary of Our Contributions:

In this paper, we present a systematic analysis of the adversarial robustness in 3D point cloud recognition using self-supervisions on three representative architectures: a multi-layer-perceptron (MLP) network (PointNet) (Qi et al., 2017), a convolutional network (DGCNN) (Wang et al., 2019), and a transformer-based network (PCT) (Guo et al., 2020). Specifically, we use two strategies to integrate self-supervised learning and adversarial training, including (1) adversarial pre-training for fine-tuning (APF), which uses the SSL tasks only for pre-training, and (2) adversarial joint training (AJT), which jointly trains the SSL task with the recognition task, as shown in Figure 1. To further study the importance of self-supervised tasks for adversarial robustness, we select three representative SSL proxy tasks, including 3D rotation prediction (Poursaeed et al., 2020), 3D jigsaw (Sauder & Sievers, 2019), and autoencoding (Yang et al., 2018). Our key observations are as follows:

- We show that pre-training on SSL tasks improves adversarial robustness of the fine-tuned models. Unlike the 2D domain, where both APF and AJT have enhanced the robustness, our study finds that only APF consistently achieves robustness improvements in 3D. AJT does not always help since the distributional gap between data for SSL and recognition tasks will distract each other in AJT. Evaluation results of various unforeseen attacks further confirm such improvements by APF.
- We find that the convolutional network, *i.e.,* DGCNN, is more robust than the other architectures. Moreover, 3D jigsaw SSL task, which predicts the permutation of 3D point cloud patches, helps achieve stronger robust-

[1] University of Michigan [2] NVIDIA. Correspondence to: Jiachen Sun <jiachens@umich.edu>.

ICML 2021 Workshop on Socially Responsible Machine Learning

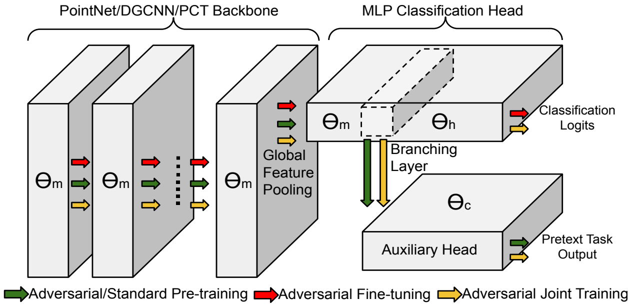

Figure 1. Overview of Our Analysis Methodology.

ness than the others. Both convolutional architecture and jigsaw SSL task enforce the model to learn better local semantics. Intuitively, robust local features help limit the propagation of adversarial effect from point-level input perturbations to the model's final output.

2. Analysis Methodology

In this section, we detail our adversarial robustness analysis methodology. We first introduce the principal 3D point cloud recognition architectures and the threat models used in our study. We then introduce two ways to generalize and improve AT using 3D point cloud SSL proxies.

2.1. 3D Point Cloud Recognition Models and Threats

We introduce the adopted model designs and the formulations of threats to 3D point clouds below.

Model Variants. We use a shared multi-layer-perceptron-based network PointNet (Qi et al., 2017), a convolutional network Dynamic Graph CNN (DGCNN) (Wang et al., 2019), and a transformer-based network Point Cloud Transformer (PCT) (Guo et al., 2020) as our primary backbone architectures, denoted as $\mathcal{M}_{\boldsymbol{\theta}_m}$. The classification head $\mathcal{H}_{\boldsymbol{\theta}_h}$ parameterized with $\boldsymbol{\theta}_h$ for these backbones is an MLP and the segmentation head $\mathcal{H}_{\boldsymbol{\theta}_h}$ is a set of 1×1 convolutions. We use $\mathcal{F}_{\boldsymbol{\theta}_f}$ parameterized with $\boldsymbol{\theta}_f$ ($\boldsymbol{\theta}_f := [\boldsymbol{\theta}_m; \boldsymbol{\theta}_h]$) to represent the overall model architecture, consisting of the stacked backbone $\mathcal{M}$ and recognition head $\mathcal{H}$, where $\mathcal{F} = \mathcal{M} \circ \mathcal{H}$. Given the input point cloud $\boldsymbol{x}$, the model $\mathcal{F}$ aims to predict the corresponding label $\boldsymbol{y}$, where $\boldsymbol{y} = \mathcal{F}(\boldsymbol{x})$.

Threats. We adopt the point shifting (PS) threat within ℓ_p projected gradient descent (PGD) style attacks. We assume a PS adversary is able to shift all existing points within a ℓ_p norm ball:

$$\boldsymbol{x}_{s+1} = \Pi_{\boldsymbol{x}+\mathcal{S}}(\boldsymbol{x}_s + \alpha \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}_s}\mathcal{L}(\boldsymbol{x}_s, \boldsymbol{y}; \mathcal{F}))) \quad (1)$$

where $\boldsymbol{x}_s$ is the adversarial example in the s-th iteration, Π is the projection function to project the adversarial example to the pre-defined perturbation space $\mathcal{S}$, and α is the attack step size.

2.2. Adversarial Training with Self-Supervisions

We first introduce the chosen 3D self-supervised learning methods, followed by two strategies to incorporate these pretext tasks in adversarial training.

3D Self-Supervised Learning. The primary goal of self-supervised learning (SSL) is to learn effective feature representations with unlabeled data. Given a pretext task P_t , the pre-training process is still conducted in a supervised manner with self-generated data $\boldsymbol{x}^t$ and label $\boldsymbol{y}^t$ from pristine data $\boldsymbol{x}$, where $(\boldsymbol{x}^t, \boldsymbol{y}^t) = P_t(\boldsymbol{x})$. Therefore, a target loss function $\mathcal{L}_t(\boldsymbol{x}^t, \boldsymbol{y}^t; \mathcal{F}^t_{\boldsymbol{\theta}_t})$ will be minimized during the optimization, where $\boldsymbol{\theta}_t$ consists of the shared backbone parameters $\boldsymbol{\theta}_m$ and customized branch parameters $\boldsymbol{\theta}_c$ (*i.e.,* $\boldsymbol{\theta}_t := [\boldsymbol{\theta}_m; \boldsymbol{\theta}_c]$). We utilize the following 3D SSL tasks in our study.

- *3D Rotation* (Poursaeed et al., 2020): Similar to the rotation task in 2D vision (Gidaris et al., 2018), the data and label are generated by rotating the original point clouds to pre-defined angles η in the 3D space. Therefore, the problem is to correctly predict 3D rotation angles *w.r.t.* the input point cloud.

- *3D Jigsaw* (Sauder & Sievers, 2019): Different from the jigsaw task in 2D vision (Noroozi & Favaro, 2016) which is defined as a classification problem, 3D jigsaw solicits a segmentation model. A point cloud is evenly divided to k^3 small cubes and shuffled to different positions. Points inside each small cube are assigned to a label signaling its original position. The problem, thus, is to correctly predict the original cube position of each point.

- *Autoencoder* (Achlioptas et al., 2018; Yang et al., 2018): An autoencoder utilizes an encoder $z = E(\boldsymbol{x})$ to learn a compact representation and a decoder $D(E(\boldsymbol{x}))$ to reconstruct the point cloud. We utilize different backbones as the encoder $E(\cdot)$ and FoldingNet (Yang et al., 2018) as the decoder $D(\cdot)$ due to its satisfactory performance. We use three different positional encodings: plane, 3D sphere, and 3D gaussian in our experiments.

Adversarial Pre-training for Fine-tuning (APF). As introduced in §1, adversarial training (AT) (Madry et al., 2017; Shafahi et al., 2019; Wong et al., 2020) has been demonstrated to be one of the most longstanding and practical defenses. We thus enable AT in both pre-training and fine-tuning stages:

$$\arg\min_{\boldsymbol{\theta}} \quad \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}\left[\max_{\sigma\in\mathbb{S}} \mathcal{L}(\boldsymbol{x}+\sigma, \boldsymbol{y}, \boldsymbol{\theta})\right] \quad (2)$$

where $\mathcal{L} \in \{\mathcal{L}_t, \mathcal{L}_f\}$ for loss functions in pre-training (t) and fine-tuning (f) stages, σ is the adversarial perturbations, and $\mathbb{S}$ represents its manipulation space. AT essentially solves a min-max problem. In the inner loop, the optimizer tries to find adversarial examples that maximize the target loss, and the outer loop updates the network parameters to correctly recognize the generated adversarial examples. In contrast, standard training (ST) is simply $\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}\left[\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right]$.

Table 1. Evaluation Results (%) of Adversarial Pre-training for Fine-tuning and Task Ensembles.

		ModelNet40						ScanObjectNN						ModelNet10					
		PointNet		DGCNN		PCT		PointNet		DGCNN		PCT		PointNet		DGCNN		PCT	
Pretext Task	Parameters	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA
AT Baseline	N/A	87.7	37.9	90.6	62.0	89.7	49.1	69.9	23.7	74.4	30.9	72.4	20.5	96.6	79.7	98.1	86.3	97.4	80.0
3D Rotation	$\eta = 6$	87.2	48.0	91.4	63.6	90.2	50.7	69.1	24.5	75.7	32.9	72.6	20.6	96.8	79.0	97.7	84.9	97.2	80.4
	$\eta = 18$	87.2	48.3	91.1	64.1	90.2	49.5	69.5	25.0	73.8	32.2	72.5	20.1	97.1	79.3	98.5	85.3	97.8	80.3
Adversarial 3D Rotation	$\eta = 6$	87.6	42.1	90.8	61.8	90.4	50.8	69.6	25.3	75.0	36.8	71.6	28.7	97.0	79.9	97.7	87.5	98.0	82.2
	$\eta = 18$	87.4	45.7	90.9	62.9	90.4	50.1	69.3	24.5	75.0	36.3	73.1	26.9	97.0	79.7	98.0	88.2	97.4	83.7
3D Jigsaw	$k = 3$	87.6	50.1	90.0	67.4	90.4	51.1	70.8	25.5	79.0	33.8	73.4	23.2	96.8	80.0	98.0	89.6	97.8	81.5
	$k = 4$	87.6	50.9	90.1	65.3	90.3	50.2	70.2	25.4	76.2	35.3	73.8	24.6	96.7	80.2	98.0	89.0	97.7	81.9
Adversarial 3D Jigsaw	$k = 3$	88.2	52.1	89.6	65.8	89.8	51.3	69.0	24.8	77.5	41.3	72.5	26.3	97.0	80.6	98.5	90.5	97.4	83.5
	$k = 4$	87.8	50.5	89.9	65.3	89.6	51.0	69.9	25.5	76.1	40.6	73.1	27.4	97.0	80.5	98.0	89.1	97.3	83.9
Autoencoder	sphere	87.4	50.0	89.9	62.8	90.2	50.7	69.9	25.1	76.1	36.0	71.3	24.1	97.0	80.5	98.2	86.8	97.1	80.1
	plane	87.1	48.8	90.1	62.2	90.2	50.2	69.4	25.5	76.2	35.6	71.1	22.6	96.8	80.8	97.8	87.6	97.0	80.1
	gaussian	87.4	48.9	90.8	63.3	89.7	50.3	69.7	23.8	75.6	35.8	71.3	24.8	96.8	80.5	97.8	86.4	97.1	80.1
Adversarial Autoencoder	sphere	87.1	49.7	90.0	62.2	90.3	50.0	70.4	25.2	75.2	36.2	72.6	22.2	96.7	80.4	97.5	87.3	97.5	82.1
	plane	86.9	46.6	89.7	61.8	89.7	50.0	69.2	24.0	75.6	38.0	73.3	21.6	97.0	80.6	98.0	86.1	97.7	82.5
	gaussian	87.1	48.5	90.7	62.7	90.2	50.5	68.8	25.0	74.7	36.3	72.6	23.4	97.0	80.2	97.8	88.4	97.4	83.2

In the pre-training stage of APF, we leverage both standard and adversarial training to get the pre-trained backbones $\mathcal{M}_{\boldsymbol{\theta}_m}$ and $\mathcal{M}^{adv}_{\boldsymbol{\theta}_m}$. Given a pre-trained backbone parameterized by $\boldsymbol{\theta}_m$, in the second stage, we adversarially fine-tune all $\boldsymbol{\theta}_f := [\boldsymbol{\theta}_m; \boldsymbol{\theta}_h]$ for the recognition task, as illustrated in Figure 1. The network branches at the penultimate vector for the rotation task and the first global feature (Sun et al., 2020b) for the jigsaw and autoencoder tasks since they use the segmentation head.

Adversarial Joint Training (AJT). Besides pre-training for fine-tuning, joint training is another way to apply SSL. The objective function is formulated as:

$$\begin{aligned} \underset{\boldsymbol{\theta}_m;\boldsymbol{\theta}_h;\boldsymbol{\theta}_c}{\arg\min} \quad & \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}\left[\max_{\sigma\in\mathbb{S}} \mathcal{L}_f(\boldsymbol{x}+\sigma, \boldsymbol{y}, \boldsymbol{\theta}_f)\right] \\ & + \lambda \cdot \mathcal{L}_t(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{\theta}_t) \end{aligned} \tag{3}$$

where λ is a hyperparameter to balance the SSL and recognition tasks. Two tasks share the same backbone $\boldsymbol{\theta}_m$ with two different branches, parameterized by $\boldsymbol{\theta}_h$ and $\boldsymbol{\theta}_c$, respectively. We also enable dual batch normalization (Xie et al., 2020) in AJT for $\boldsymbol{x}$ and $\boldsymbol{x}_t$ since they should belong to different underlying distributions. We use two model-agnostic tasks, *i.e.,* 3D rotation and jigsaw in AJT.

Similarly, in our AJT analysis, all $\mathcal{L}_t$ and $\mathcal{L}_f$ can be formulated as cross-entropy loss. We set $\lambda = 1$ and leverage the same branching point with APF. The whole network is trained to predict the supervised task with the original head and the SSL task with the auxiliary head (Figure 1).

3. Experiments and Results

We here present our experimental setups and results.

3.1. Evaluation Setups

Datasets. We leverage three datasets ($\mathcal{D}$): ModelNet40 (Wu et al., 2015) (40 classes), ModelNet10 (Wu et al., 2015) (10 classes), and ScanObjectNN (Uy et al., 2019) (15 classes) throughout our experiments. For each point cloud, we randomly sample 1024 points and normalize it to an edge-length-2 cube ($[-1, 1]$) for experimentation. For SSL, we randomly sample $\boldsymbol{y}^t$ from the pre-defined label sets and further generate $\boldsymbol{x}^t$ based on $\boldsymbol{y}^t$ in each iteration. Specifically, we choose $\eta = 6, 18$ and $k = 3, 4$ for rotation and jigsaw tasks, followed by the suggestion of Poursaeed et al. (2020) and Sauder & Sievers (2019).

Adversary. As introduced in §2.1, we exploit 7-step and 200-step ℓ_∞ PGD attacks (Madry et al., 2017) targeting the cross-entropy loss for adversarial training and testing, respectively. We follow Sun et al. (2020b) to empirically set the perturbation boundary $\epsilon = 0.05$ ($||\sigma||_\infty \leq 0.05$). We utilize PGD step size $\alpha = 0.01$ and $\alpha = 0.005$ in the training and testing phases, respectively.

Training Details. All pre-trained and fine-tuned models in APF are trained using Adam (Kingma & Ba, 2014). We use batch sizes of 32 for PointNet and DGCNN, and 128 for PCT. The initial learning rate is set to 0.001 for PointNet and DGCNN, and 5×10^{-4} for PCT. Both pre-training and fine-tuning take 250 epochs, where a 10× decay happens at the 100-th, 150-th, and 200-th epoch. We leverage the same training setups in AJT. All experiments are done on 1 to 4 NVIDIA V100 GPUs (v10, 2020).

3.2. Self-Supervised Pre-training Helps Adversarial Fine-tuning

We systematically evaluate all configurations in APF under PS attack. As introduced earlier, we use standard and adversarial training to get the pre-trained models. From Table 1[1], we can make several interesting observations. First, we find that our APF strategy generally enhances the adversarial robustness. The best-fine-tuned models achieve 14.2%, 5.4%, and 2.2% robustness improvements in PointNet, DGCNN, and PCT on ModelNet40, respectively. The enhancements on the real-world dataset, ScanObjectNN, *i.e.,* 1.8%, 10.4%, and 6.9% in PointNet, DGCNN, and PCT, are also signifi-

[1] The 1-st and 2-nd highest accuracy among fine-tuned models in each column are noted, and we use the same mark throughout this paper.

Table 2. Evaluation Results (%) of Adversarial Joint Training.

		ModelNet40						ScanObjectNN						ModelNet10					
		PointNet		DGCNN		PCT		PointNet		DGCNN		PCT		PointNet		DGCNN		PCT	
Pretext Task	Parameters	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA	CA	RA
AT Baseline	N/A	87.7	37.9	90.6	62.0	89.7	49.1	69.9	23.7	74.4	30.9	72.4	20.5	96.6	79.7	98.1	86.3	97.4	80.0
3D Rotation	$\eta = 6$	86.8	45.0	91.2	60.7	89.5	44.3	67.8	24.3	74.2	37.8	72.3	20.3	96.6	79.0	98.1	86.3	97.8	73.8
	$\eta = 18$	86.5	46.4	91.3	62.0	88.9	42.9	68.7	25.1	76.2	37.2	72.1	19.8	97.0	79.9	97.9	85.7	98.1	75.6
3D Jigsaw	$k = 3$	87.6	42.5	91.0	62.3	90.2	43.1	69.4	25.5	77.1	38.9	72.1	20.7	96.8	79.8	98.4	87.9	97.7	76.8
	$k = 4$	87.2	46.7	91.1	61.7	89.8	40.9	70.0	24.6	75.9	38.4	73.7	20.8	96.8	77.9	98.0	88.6	97.1	78.0

cant, which demonstrate the generality of APF. Second, we find that DGCNN outstands to be the most robust architecture, consistently achieving $\sim$15% stronger robustness than the other two models on both ModelNet40 and ScanObjectNN. Lastly, jigsaw-based APF offers more robustness improvements than the other two methods while maintaining slightly higher clean accuracy (CA). Specifically, jigsaw-based APF, on average, further boosts DGCNN's robust accuracy (RA) by 2.8%, 2.2%, and 2.7% on three datasets, respectively.

Insights. Different from 2D images that possess both texture and shape information, 3D point clouds naturally bias towards shape. In 2D image space, it is widely recognized that local and global features correspond to the texture and shape information, respectively (Bui et al., 2020). Recent studies have demonstrated that appreciation of global/shape features can help improve model robustness on image classification (Chalasani et al., 2018). However, we find some distinctions in point cloud recognition. As mentioned above, PointNet with only global feature learning will be easily affected by the perturbed points (Table 1). Due to the sparsity of point clouds, the local feature actually represents the smoothness of the object's surface. Thus, learning robust local features is critical for correctly recognizing a perturbed point cloud, as it limits the adversarial effect propagation to the model output.

As summarized above, DGCNN achieves the strongest robustness under AT, attributed to the hierarchical usage of EdgeConv (Wang et al., 2019). EdgeConv dynamically aggregates local features by exploiting kNN. Such an aggregation method has the ability to calibrate the adversarial effect in the local feature learning stage. Although transformer-based architectures have gained tremendous visibility recently (Dosovitskiy et al., 2020), we find that PCT does not have a major robustness improvement compared to PointNet. Self-attention increases the capacity of the model architecture, but it also enlarges the receptive field of the model (Zhang et al., 2019). In PCT, each point can influence every other point's feature, which will potentially increase the model's fragility (Xiang et al., 2020).

Moreover, we also find that jigsaw-based APF is the most effective method to improve adversarial robustness, aligning well with our above insights. Jigsaw SSL makes the model learn to reassemble the randomly displaced local point clusters, where the model is enforced to learn the displaced local features. Meanwhile, to correctly reconstruct the point cloud, jigsaw SSL also requires the model to capture the global and holistic semantics. Nevertheless, rotation and autoencoder-based pre-training methods focus more on global feature learning. Therefore, we believe jigsaw-based APF is a perfect candidate to strengthen the association between local and global features in point cloud learning, hence improving the adversarial robustness under APF.

3.3. Adversarial Joint Training Does not Always Improve Robustness

We leverage two model-agnostic SSL tasks in AJT. As presented in Table 2, AJT can still enhance the robustness in PointNet and DGCNN. For instance, AJT improves their RA by 1.8% and 8.0% on ScanObjectNN, respectively. However, AJT overall cannot outperform APF in point cloud recognition. Especially, we find AJT even degrades the RA of PCT compared to the standard AT.

Insights. We find this also to be related to the natural characteristic of point cloud data. Although SSL can help models learn strong priors and context information, it is still a *separate* learning task. Rotated and disassembled images still preserve similar local features to the original images since the RGB values do not change, so that the auxiliary optimization in AJT will not distract AT but help models learn robust global features (Hendrycks et al., 2019). However, point cloud models take point coordinates xyz as input. Rotated and disassembled point clouds have significant variations in their coordinates' numeric values. Although we apply dual batch normalization (Xie & Yuille, 2020) to migrate the feature heterogeneous problems, such discrimination will consequently distract model learning in AJT, and thus hurt the RA performance. The usage of self-attention in PCT will further expand this impact since it introduces a global receptive field (Xiang et al., 2020).

4. Conclusion

In this work, we systematically explore the impact of self-supervised learning (SSL) on the adversarial robustness in 3D point cloud recognition. We find tangible robustness improvements by the adversarial pre-training for fine-tuning strategy. We also experimentally show that robust local features are critical to achieving robustness in 3D, explaining the success of DGCNN and the jigsaw proxy task. Our results shed light for future research on designing more robust models and SSL schemes for 3D point clouds.

References

NVIDIA V100 TENSOR CORE GPU. https://www.nvidia.com/en-us/data-center/v100/, 2020.

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pp. 40–49. PMLR, 2018.

Bui, A., Le, T., Zhao, H., Montague, P., deVel, O., Abraham, T., and Phung, D. Improving adversarial robustness by enforcing local and global compactness. *arXiv preprint arXiv:2007.05123*, 2020.

Cao, Y., Xiao, C., Cyr, B., Zhou, Y., Park, W., Rampazzi, S., Chen, Q. A., Fu, K., and Mao, Z. M. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pp. 2267–2281, 2019.

Chalasani, P., Chen, J., Chowdhury, A. R., Jha, S., and Wu, X. Concise explanations of neural networks using adversarial training. *arXiv*, pp. arXiv–1810, 2018.

Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., and Wang, Z. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 699–708, 2020.

Choy, C., Gwak, J., and Savarese, S. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., and Hu, S.-M. Pct: Point cloud transformer. *arXiv preprint arXiv:2012.09688*, 2020.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.

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

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Maturana, D. and Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928. IEEE, 2015.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Poursaeed, O., Jiang, T., Qiao, Q., Xu, N., and Kim, V. G. Self-supervised learning of point clouds via orientation estimation. *arXiv preprint arXiv:2008.00305*, 2020.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Qi, C. R., Zhou, Y., Najibi, M., Sun, P., Vo, K., Deng, B., and Anguelov, D. Offboard 3d object detection from point cloud sequences. *arXiv preprint arXiv:2103.05073*, 2021.

Riegler, G., Ulusoy, A. O., and Geiger, A. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Sauder, J. and Sievers, B. Self-supervised deep learning on point clouds by reconstructing space. *arXiv preprint arXiv:1901.08396*, 2019.

Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

Shi, S., Wang, X., and Li, H. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–779, 2019.

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., and Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020.

Sun, J., Cao, Y., Chen, Q. A., and Mao, Z. M. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 877–894. USENIX Association, August 2020a. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/sun.

Sun, J., Koenig, K., Cao, Y., Chen, Q. A., and Mao, Z. M. On the adversarial robustness of 3d point cloud classification, 2020b.

Uy, M. A., Pham, Q.-H., Hua, B.-S., Nguyen, T., and Yeung, S.-K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1588–1597, 2019.

Wang, D. Z. and Posner, I. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, volume 1, pp. 10–15607. Rome, Italy, 2015.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38 (5):1–12, 2019.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Xiang, C., Qi, C. R., and Li, B. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9136–9144, 2019.

Xiang, C., Bhagoji, A. N., Sehwag, V., and Mittal, P. Patchguard: Provable defense against adversarial patches using masks on small receptive fields. *arXiv preprint arXiv:2005.10884*, 2020.

Xie, C. and Yuille, A. Intriguing properties of adversarial training at scale. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HyxJhCEFDS.

Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A. L., and Le, Q. V. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 819–828, 2020.

Yang, Y., Feng, C., Shen, Y., and Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 206–215, 2018.

Yin, T., Zhou, X., and Krähenbühl, P. Center-based 3d object detection and tracking. *CVPR*, 2021.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention generative adversarial networks. In *International conference on machine learning*, pp. 7354–7363. PMLR, 2019.