EXTENDING ADVERSARIAL ATTACKS AND DEFENSES TO DEEP 3D POINT CLOUD CLASSIFIERS

Daniel Liu[⋆] *Ronald Yu*[†] *Hao Su*[†]

[⋆] Torrey Pines High School, San Diego, CA
[†] University of California, San Diego, La Jolla, CA

ABSTRACT

3D object classification using deep neural networks has been extremely successful. As the problem of identifying 3D objects has many safety-critical applications, the neural networks have to be robust against adversarial changes to the input data set. We present a preliminary evaluation of adversarial attacks on 3D point cloud classifiers by evaluating adversarial attacks that were proposed for 2D images, and extending those attacks to reduce the perceptibility of the perturbations in 3D space. We also show the effectiveness of simple defenses against those attacks. Finally, we attempt to explain the effectiveness of the defenses through the intrinsic structures of both the point clouds and the neural networks. Overall, we find that 3D point cloud classifiers are weak to adversarial attacks, but they are also more easily defensible compared to 2D image classifiers. Our investigation will provide the groundwork for future studies on improving the robustness of deep neural networks that handle 3D data.

***Index Terms*—** adversarial attack, adversarial defense, 3D point cloud, deep neural network, fast gradient method

1. INTRODUCTION

Recent advances in 3D deep learning have made strides in tasks previously established by 2D baselines such as classification [1], segmentation [2], and detection [3]. Much of the research in 3D deep learning has been on processing various representations of 3D objects, like point clouds [4, 5, 1], meshes [6], and voxels [7]. However, 3D deep learning literature still lags behind its 2D counterpart on tasks that seek to better understand behavior of deep neural networks such as network interpretation [8], few-shot learning [9], and robustness to adversarial examples [10]. We provide a preliminary investigation into how deep 3D neural networks behave by examining the behavior of 3D point cloud classifiers on adversarial attacks that are extremely effective on 2D images.

Robustness against adversarial attacks has been subject to rigorous research due to its security implications in deep learning systems. Deep neural networks were shown to be extremely vulnerable against adversarial perturbations [11, 10, 12, 13, 14, 15, 16, 17, 18], which suggests a fundamental limitation in neural networks. Also, there have been many techniques proposed for defense, including adversarial training [10] and defensive distillation [19]. However, so far, there are no universal defenses that are effective against all adversarial attacks. This indicates the difficulty of the challenge posed by adversarial attacks [20]. Furthermore, the perturbations generated by adversarial attacks are relatively *imperceptible* to humans, yet extremely effective in fooling neural networks. Only very recently has there been research on adversarial attacks and defenses for 3D point clouds [21, 22, 23].

We seek to advance studies in both 3D point cloud classification and adversarial robustness by examining the robustness of point cloud classifiers, namely PointNet and PointNet++. We show that various white-box adversarial attacks are effective on undefended deep point cloud classifiers, and propose effective defenses against those attacks. We find that deep 3D point cloud classifiers are still susceptible to simple adversarial attacks, but they are also be more easily defensible than their 2D counterparts against some white-box attacks, due to intrinsic properties of the models and the point cloud data.

2. WHITE-BOX ADVERSARIAL ATTACKS

As a preliminary investigation on the robustness of 3D deep neural networks to adversarial examples, we explore untargeted (*i.e.*, misclassify to any class other than correct label) variations of the fast gradient method introduced by Goodfellow *et al.* [10], which, despite its simplicity, was shown to be highly effective on 2D images.

2.1. Fast gradient method

The fast gradient sign method (FGSM) introduced by Goodfellow *et al.* [10] generates adversarial examples against a deep neural network f (that is parameterized by θ and takes an input x) by increasing its cross entropy loss J between the network's output and the label y while constraining the L_∞ norm of the

A preprint of this paper is available at this http url: https://arxiv.org/abs/1901.03006

perturbation of x:

$$x^{adv} = x + \epsilon sign(\Delta_x J(f(x;\theta), y)) \quad (1)$$

The ϵ value is an adjustable hyperparameter that dictates the L_∞ norm of the difference between the original input and the adversarial example.

The iterative fast gradient method [12] improves the fast gradient attack by repeating it multiple times to get a better estimate of the loss surface wrt to the input of the network.

2.2. Modifying the fast/iterative gradient method

We expand Goodfellow *et al.*'s [10] idea to several related categories of attacks. All of these cases constrain the magnitude of the perturbation onto the surface of an L_2 ϵ-ball, but in different dimensions.

- Constraining the perturbation for each dimension onto the surface of a 1D L_2 epsilon ball (essentially $||x^{adv} - x||_\infty \leq \epsilon$). This is just Goodfellow *et al.*'s FGSM [10].
- Constraining the L_2 norm of the perturbation for each point onto the surface of a 3D L_2 epsilon ball ($||p^{adv} - p||_2 \leq \epsilon, \forall p \in x, p^{adv} \in x^{adv}$). We refer to this as the "normalized gradient L_2 method".
- Constraining the L_2 norm between the entire clean point cloud and the entire adversarial point cloud [24, 25] ($||x^{adv} - x||_2 \leq \epsilon$). This allows the individual perturbations to have diverse magnitudes and directions. We refer to this as the "gradient L_2 method".

Our preliminary tests have shown little difference between the iterative attack success rates of all three methods. However, we mainly consider the latter two attacks due to the severely limited number of directions of perturbations generated by fast gradient sign.

2.3. Other approaches

One main problem with using adversarial attacks in 3D space is that, unlike 2D space, the perturbations are more perceptible due to obvious outliers. As such, in addition to those basic attacks, we also propose methods that reduce the perceptibility of those attacks.

2.3.1. Gradient projection

In this method, perturbations are orthogonally projected onto the surface of an object's unperturbed shape, which is made up of a mesh of triangles. For each perturbed point p^{adv}, we project it onto the plane of the triangle it was sampled from:

$$p^{adv}_{proj} = p^{adv} - n[n \cdot (p^{adv} - t_1)] \quad (2)$$

where n is the unit normal vector of the plane and t_1 is a vertex of the triangle. Then, we clip the point to the edges of the triangle if it leaves the triangle.

This method shows that we can generate adversarial attacks of a point cloud by simply changing the sampling density, and it allows us to measure how the networks perform against changes to the distribution of points. Furthermore, it generates adversarial examples that are barely perceptible to humans. The drawback of this approach is that the triangular mesh of each object is needed, which is not available for point cloud data obtained in practice.

2.3.2. Clipping norms

A more practical way to lower the perceptibility of attacks is to clip the L_2 norms of the perturbation of each point in order to match the mean pairwise euclidean distances between nearby points in the clean sample. This limits large, outlying perturbations that may occur in one of the basic attacks.

3. DEFENSES

We evaluate the performance of several simple defensive techniques in response to the adversarial attacks. In addition to evaluating adversarial training, we also propose two different input restoration methods that try to remove perturbed points by making certain assumptions about clean input point clouds.

3.1. Adversarial training

The adversarial training algorithm was initially proposed by Goodfellow *et al.* [10]. We train each model from scratch by, generating adversarial examples at each iteration and averaging the loss from feeding in batches of clean and adversarial examples.

3.2. Input restoration

3.2.1. Removing outliers

Another way to defend against adversarial attacks is by removing outlying points that may be created due to adversarial perturbations. This is similar to ideas from [22].

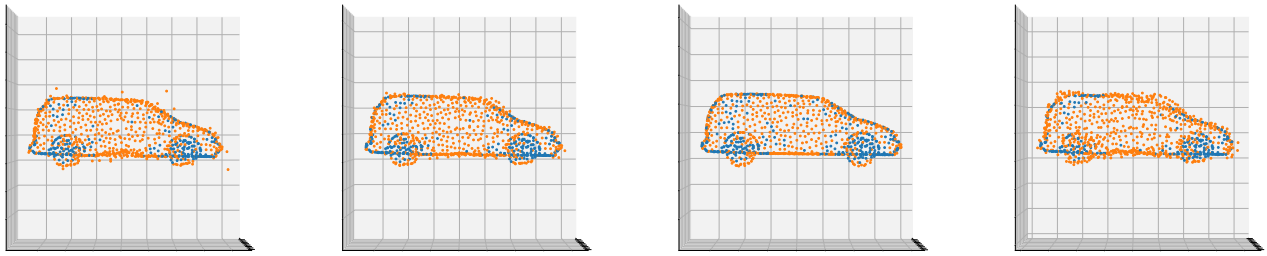
Outliers are identified by examining the mean euclidean distance of each point to its k-nearest neighbors. Points that have very high mean distances are assumed to be outliers and are discarded. This method assumes that since each point on a natural shape should be uniformly sampled along the surface, any outlier point must be the result of adversarial perturbations.

3.2.2. Removing salient points

We also explore a defensive technique supported by the crude assumption that perturbed points should have relatively large magnitudes of gradients. By assuming that this is true, an

	None	Adversarial training	Removing outliers	Removing salient points
None	0.0%	0.5%	2.6%	0.7%
Fast gradient L_2	39.8%	7.3%	4.6%	10.2%
Iter. gradient L_2	74.2%	37.1%	16.2%	19.9%
Iter. gradient L_2, clip norm	45.2%	32.6%	10.9%	14.2%
Iter. gradient L_2, grad. proj.	4.3%	6.8%	2.5%	2.1%
Normalized fast gradient L_2	14.1%	7.9%	9.1%	12.9%
Normalized iter. gradient L_2	64.5%	59.1%	19.7%	32.2%

Table 1: Success rates of untargeted attacks on PointNet trained with ModelNet-Unique, under different defense methods. Each column represents a defense method, and each row represents an attack method.



(a) Iter. gradient L_2

(b) Iter. gradient L_2 and clipping norms

(c) Iter. gradient L_2 and gradient proj.

(d) Normalized iter. gradient L_2

Fig. 1: A set of successful adversarial perturbations on a car. The attacks were performed on PointNet trained with the ModelNet-Unique dataset. All of them were misclassified as range hoods. Orange points have nonzero perturbations.

algorithm that discards salient points can be used. The saliency of each point p^{adv} is given by:

$$\max_i \left|\left| \frac{\partial f_i(x^{adv};\theta)}{\partial p^{adv}} \right|\right|_2 \tag{3}$$

where $f_i(\cdot;\theta)$ represents the value of the i-th output class of the model f.

4. EVALUATION

4.1. Models

We evaluate both PointNet [1] and PointNet++ [5] for their performance against the mentioned adversarial attacks and defenses. We directly use the default hyperparameters when training the networks, except for a slightly lower batch size for PointNet++ due to limited memory.

4.2. Datasets

We use shapes from the ModelNet-40 [26] dataset to train and evaluate the models. There are over 2400 total objects from 40 different classes in the dataset.

We sample 1024 points from each object and we center and scale the data to match the settings used by Qi *et al.* [1, 5] for PointNet and PointNet++.

Since some of the classes in ModelNet-40 are quite indistinguishable even to humans (*e.g.*chair and stool), for most experiments, we use a subset of 16 hand-picked object classes that have more unique shapes, which allows us to measure the effectiveness of adversarial attacks that have to switch between very different classes. The 16 classes are: airplane, bed, bookshelf, car, chair, cone, cup, guitar, lamp, laptop, person, piano, plant, range hood, stairs, and table. We will refer to this dataset as ModelNet-Unique.

4.3. Implementation details

For all attacks that constrain the L_2 norm between the clean and adversarial point clouds, we use an ϵ value of 1. For normalized fast/iterative gradient attacks, we use an ϵ value of 0.05. We run all iterative attacks for 10 iterations.

For our evaluations of the defensive techniques, we adversarially train with perturbations generated by fast gradient L_2 using an ϵ value of 1. For the outlier removal method, we use the mean distance to the 10 closest neighbors of each point and we clip perturbations that exceed the mean by 1 standard deviation. We remove 100 of the most salient points when removing salient points.

5. RESULTS

5.1. Clean inputs

We perform all of our attacks on only the correctly classified point clouds. For PointNet and PointNet++ using the full 40 classes, around 90% of the point clouds are correctly classified. On ModelNet-Unique, around 96% of the point clouds are correctly classified.

5.2. Effectiveness of white-box attacks and defenses

	Success rate
Fast gradient L_2	58.8%
Iter. gradient L_2	90.1%
Iter. gradient L_2, clip norm	77.0%
Iter. gradient L_2, gradient proj.	26.0%
Normalized fast gradient L_2	40.0%
Normalized iter. gradient L_2	88.1%

Table 2: Success rates for untargeted attacks on PointNet trained with ModelNet-40.

	40	Unique
Fast gradient L_2	36.5%	36.1%
Iter. gradient L_2	96.4%	92.2%
Iter. gradient L_2, clip norm	91.2%	70.6%
Iter. gradient L_2, grad. proj.	24.5%	4.6%
Normalized fast gradient L_2	31.0%	24.7%
Normalized iter. gradient L_2	96.6%	91.6%

Table 3: Success rates of untargeted attacks on PointNet++. The network is trained/evaluated on both ModelNet-40 (40) and ModelNet-Unique (Unique).

Adversarial attacks on undefended PointNet and PointNet++ networks are extremely effective (>90% attack success rate with iterative gradient attacks on both networks trained with ModelNet-40). Furthermore, PointNet++ has higher error rates than PointNet for both the vanilla and the normalized versions of the iterative gradient L_2 attack, even though it is more complex, which suggests that higher architecture complexity does not lead to higher robustness against adversarial attacks.

The defenses we evaluate are also effective. For iterative gradient L_2 that has a success rate of 74.2% on PointNet trained with ModelNet-Unique, adversarial training lowers the success rate to 37.1%, removing salient points lowers it to 19.9%, and removing outliers performs even better by lowering it to 16.2%. However, adversarial training is much less effective against normalized iterative gradient L_2 compared to iterative gradient L_2. This suggests that it does not transfer very well to perturbations that have different distributions.

Overall, the two input restoration defenses perform even better than adversarial training. We find that both removing outliers and removing salient points, which were constructed to defend against large perturbations, are also effective against attacks like L_2 norm clipping and gradient projection that generate small perturbations. Furthermore, directly removing salient points does not damage the classification of clean input point clouds by too much compared to other methods.

The iterative gradient L_2 attack with gradient projection is the least perceptible attack. However, It reaches over 20% success rate on both PointNet and PointNet++ with the ModelNet-40 dataset, even though there are barely any visible changes to the input point clouds. With higher epsilons, it attains over 30% to 40% success rate on both networks.

6. DISCUSSION

PointNet and PointNet++ have been shown to be robust against point clouds of varying densities and randomly perturbed point clouds [1, 5]. However, against our adversarial attacks that preserve the overall shape of the input point clouds by attempting to minimize human-perceptibility, the networks perform very poorly.

However, the last max-pooling layer in PointNet and PointNet++ allows input restoration defenses to work well against adversarial attacks. It allows many redundant, "non-critical" points to not have any gradients since they are not selected by the max-pooling operation. This means that those points cannot be perturbed by white-box attacks that require gradients for each point. As outlier points or salient points are removed, those non-critical points are then able to represent the overall shape of the input point cloud and allow the networks to get accurate prediction. In a sense, the true shape of a point cloud is hidden inside the perturbed version of the point cloud. This can be empirically observed by looking at the blue points in Figure 1, which have zero gradients and therefore, are not perturbed. This property allows 3D point cloud classifiers like PointNet and PointNet++ to be more easily defensible than 2D image classifiers against attacks that require gradients.

We think that our outlier removal method provides a necessary upper bound for future evaluations of adversarial attacks in 3D space, as unlike image pixels, each point can be perturbed by an arbitrary amount. Removing very obvious outliers is necessary to prevent attacks on 3D point clouds that may create effective, but unrealistic changes to the input data.

7. CONCLUSION

As deep neural networks are applied to various problems, the significance of adversarial examples grows. We hope that our work can provide a foundation for further research into improving the robustness of neural networks that handle 3D data in safety-critical applications.

8. REFERENCES

[1] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, pp. 4, 2017.

[2] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas, "Frustum PointNets for 3D object detection from RGB-D data," *arXiv preprint arXiv:1711.08488*, 2017.

[4] Haowen Deng, Tolga Birdal, and Slobodan Ilic, "PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors," *arXiv preprint arXiv:1808.10322*, 2018.

[5] 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*, 2017, pp. 5099–5108.

[6] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas, "SyncSpecCNN: Synchronized spectral cnn for 3D shape segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6584–6592.

[7] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 72, 2017.

[8] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba, "Interpreting deep visual representations via network dissection," *arXiv preprint arXiv:1711.05611*, 2017.

[9] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[10] IJ Goodfellow, J Shlens, and C Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[11] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[13] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li, "Boosting adversarial attacks with momentum," *arXiv preprint*, 2018.

[14] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy*. IEEE, 2016, pp. 372–387.

[15] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[16] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy*. IEEE, 2017, pp. 39–57.

[17] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli, "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 387–402.

[18] Anurag Arnab, Ondrej Miksik, and Philip HS Torr, "On the robustness of semantic segmentation models to adversarial attacks," *arXiv preprint arXiv:1711.09856*, 2017.

[19] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[20] Nicholas Carlini and David Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 3–14.

[21] Chong Xiang, Charles R Qi, and Bo Li, "Generating 3D adversarial point clouds," *arXiv preprint arXiv:1809.07016*, 2018.

[22] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu, "Deflecting 3D adversarial point clouds through outlier-guided removal," *arXiv preprint arXiv:1812.11017*, 2018.

[23] Tianhang Zheng, Changyou Chen, Kui Ren, et al., "Learning saliency maps for adversarial point-cloud generation," *arXiv preprint arXiv:1812.01687*, 2018.

[24] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[25] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii, "Distributional smoothing with virtual adversarial training," *arXiv preprint arXiv:1507.00677*, 2015.

[26] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.