Evaluating Corruption and Adversarial Robustness of Point Cloud Classifiers

Masa Nakura
University of Washington
mnakura@uw.edu

Chung Yik Edward Yeung
University of Washington
chungy04@uw.edu

Abstract

Point Clouds data has been increasingly used for various safety-critical computer vision applications such as driver assistance systems. Thus, its robustness against both adversarial attacks and corrupted data is crucial. This paper investigates the relationship between the two robustness in point cloud classifier models like PointNet and Dynamic Graph CNN(DGCNN). In the process, we analyze the relationship between corruption and adversarial robustness to establish a stronger connection between the two robustness metrics than it has been studied before. Next, our key contribution is the development and validation of a new adversarial training method using Auto-PGD attacks, which diverges from the traditional PGD approach. Our findings indicate that APGD has the potential to achieve both improved corruption and adversarial robustness. Through our analysis, we encourage further research in APGD-based adversarial training and highlight the importance of simultaneously addressing both corruption and adversarial robustness, thus paving the way for more reliable point cloud data analysis in real-world applications.

1. Introduction

1.1. Background

Point cloud data, characterized by an unordered list of points, plays a vital role in numerous vision-based applications like autonomous driving and robotics. Further developments in classifiers for point cloud data is crucial, especially in safety-critical applications such as advanced driver assistance systems (ADAS) where the model distinguishes pedestrians from vehicles.

Recent advances in point cloud processing, particularly models like PointNet [3] and its successors, PointNet++ [2] and PointNeXt [9], have enabled the direct processing of point clouds, thereby bypassing the need for intensive preprocessing like voxelization. This evolution marked a significant leap in handling unstructured point cloud data efficiently. Moreover, the adaptation of Transformer models, primarily known for their success in Natural Language Processing (NLP) and Computer Vision (CV), to point cloud data [6], has opened new avenues in point cloud analysis.

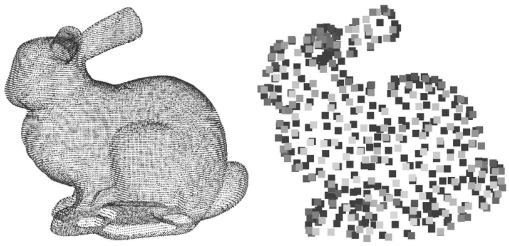

Figure 1. Point cloud object bunny (left) and voxelized object bunny (right) [1]

Despite these advancements, the integrity of point cloud classifiers in the face of data corruption and adversarial attacks is still an open question. A comprehensive survey paper by Sun et al. [17] highlighted the potential of adversarial training, particularly using Projected Gradient Descent (PGD), in enhancing the robustness against corrupted data. They found that while adversarial training improved resilience against data similar to adversarial examples, it did not conclusively outperform other data augmentation techniques in improving corruption robustness. This calls for further exploration into methods that simultaneously boost both adversarial and corruption robustness in point cloud classifiers.

1.2. Dataset

ModelNet40, part of the larger ModelNet collection [21], includes 3D computer-aided design (CAD) models across 40 object categories. In our study, we utilize ModelNet40-C for testing and training, a specialized version of ModelNet40 designed to assess the robustness of 3D point cloud recognition systems against various corruptions. ModelNet40-C [17], developed by Sun et al. [17], extends ModelNet40 by introducing 15 types of corruption across 5 severity levels, including issues related to density, noise, and transformation. The ModelNet40-C dataset comprises 185,000 unique point clouds, offering a comprehen-

sive view of model performance under various corrupted conditions. See Fig. 2 for the types of corruptions included in this dataset.

While ModelNet40-C is used to evaluate corruption robustness, we will use the ModelNet40 dataset for evaluating adversarial robustness. By using the clean ModelNet40 dataset, we generate adversarial examples, which we then can use to train or test our models.

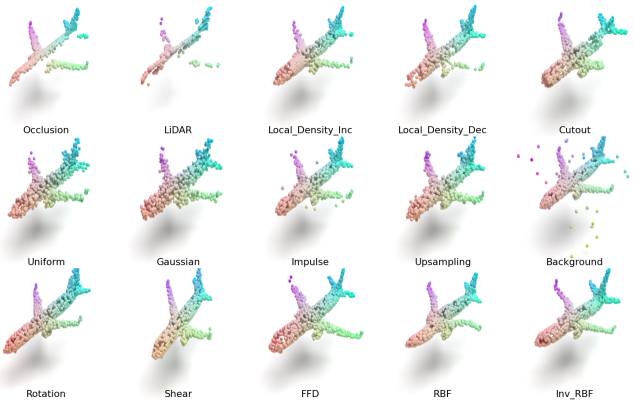

Figure 2. 3D point cloud model of an airplane subjected to various types of corruptions pulled from the ModelNet40-C dataset [17]

1.3. Contributions

Our research aims to take a step towards discovering a methodology to improve the corruption and adversarial robustness of point cloud models. In our study, we first revisit and deepen our understanding of the relationship between corruption robustness and adversarial robustness in point cloud models. By investing these aspects, our study seeks to provide new insights into the underlying dynamics governing robustness in point cloud classifiers. We then explore the potential of Auto-PGD(APGD) attacks as a method for defense through adversarial training, which diverges from the traditional method using PGD.

Specifically, our contributions are as follows:

- Offering detailed insight into the relationship between corruption and adversarial robustness.
- Demonstrating the potential of APGD by training our own DGCNN model, which outperformed all other pre-trained DGCNN models with various data-augmentation techniques.

2. Related Works

2.1. Point Clouds

Point Cloud data represents three-dimensional geometric information, typically generated via 3D scanning technologies. This data is crucial for various domains like computational design, robot navigation, and driving assistance systems. Deep Neural Networks (DNNs) are employed to perform tasks like classification, object detection, and semantic segmentation on point clouds. Previously, handling point clouds with DNNs was challenging due to their unstructured nature, often requiring voxelization for data preprocessing. However, introduced in 2016, PointNet [3] enabled direct processing of point clouds without voxelization, thus eliminating the need for heavy pre-processing. Following PointNet, models like PointNet++ [2] and PointNeXt [9] emerged, improving upon the original model by introducing hierarchical network structures and refined training strategies.

Recently, more frameworks like Point Cloud Transformers [6] have developed, applying Transformer models, initially used in Natural Language Processing (NLP). On another front, PointMLP [15] also have been introduced as a more straightforward way to classify point clouds by reducing the need for extracting local geometries.

2.2. Point Cloud Corruption

Significant research regarding corrupted point clouds has been conducted by Li et al. [14], which addresses the issue that LiDAR detectors are vulnerable to real-world corruptions like rain, snow, and sensor noise despite high accuracy in standardized benchmarks. To tackle this, the authors propose physical-aware simulation methods to generate degraded point clouds under various real-world corruptions. They construct a benchmark containing over 1.1 million examples covering 7,481 scenes, 25 corruption types, and 6 severities.

More recently, ModelNet40-C [17] has been developed as another way to benchmark corruption robustness by providing a complete dataset of corrupted point clouds, as we have described earlier. The authors of ModelNet40-C have also publicized all of their pre-trained models that were used in their study to evaluate corruption robustness. By doing so, they encourage external researchers to leverage their dataset and pre-trained models in order to further study corruption robustness.

2.3. Adversarial Attacks and Adversarial Training

One of the most common methods to attack a model is the Projected Gradient Descent (PGD), which was introduced by Madry et al [13]. Evasion attacks like PGD purposefully perturbs existing data in order to create a data point that resembles the original data but is misclassified by the model. In face of these threats, adversarial training is a crucial method for assessing the vulnerabilities of deep learning algorithms. This approach has its roots in the same paper by Madry et al [13], where a model is trained using perturbed data instead of the original, clean data.

Defense algorithms such as adversarial training has been an increasing research topic in recent years, which calls for the need for a standardized method of evaluating adversarial robustness metrics. Croce and Hein introduced a tool called AutoAttack [5], which provides a variety of attacks that can be used to benchmark a model's robustness to attacks. These attacks in AutoAttack require no parameters to run, removing any biases induced by human evaluation methods. Thus, it provides a universal method to test any model for adversarial robustness.
Furthermore, AutoAttack introduces Auto-PGD (APGD) as one of its attack methodologies. While APGD provides a more rigorous method to find an adversarial example compared to PGD, it has not been used in the context of adversarial training to the best of our knowledge. Thus, we plan to explore the use of APGD as a method of adversarial training in the context of point cloud classifiers through our work.

While many research efforts in adversarial attacks and defense focus on image classification, such as in AutoAttack, there have been efforts to investigate adversarial robustness in the domain of point cloud classifiers as well. The study conducted by Naderi et al. [10] provides a broad survey on attacks and defenses for point cloud classifiers, which provides us a good foundation for our study.

3. Methods

In this section, we discuss our methods of achieving our two goals: evaluating the correlation of adversarial robustness with corruption robustness and using Auto-PGD to adversarially train a point cloud classifier method. We will do do so by leveraging the frameworks provided by ModelNet40-C [17] and AutoAttack [5]. ModelNet40-C contains a zoo of pre-trained point cloud classifiers such as PointNet [3], PointNet++ [2], DGCNN [18], RSCNN [12], PCT [8], and SimplView [7] with various data augmentation strategies including PointCutMix-R, PointCutMix-K [20], PointMixup [4], RSMix [11], and Projected Gradient Descent(PGD) [16]. We specifically evaluate DGCNN and PointNet with various data augmentation in our study in the interest of time. Given these models, we use AutoAttack to evaluate the adversarial robustness and compare its results with the corruption robustness evaluated through ModelNet40c. Then we will use Auto-PGD(APGD), introduced by AutoAttack, as an alternative adversarial training strategy to train these point cloud classifers in contrast to the traditional PGD method.

3.1. Adversarial Attacks

In order to attack a model, we first need to specify a threat model, also known as an attack model. A threat model specifies the type of attack an attacker uses to purposefully mis-classify a data point x. Precisely, we define

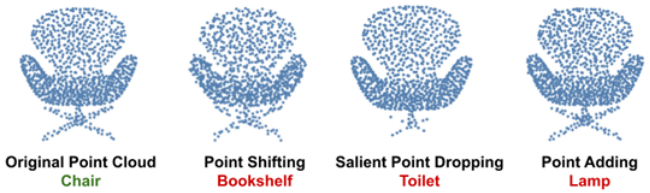

Figure 3. Adversarial examples of a chair using PGD [16]

the allowed perturbation on a given input point x. Commonly, the allowed perturbation is the space in the ℓ_∞ ball x with a maximum perturbation distance of ϵ. We choose ϵ such that all perturbed x inside the ℓ_∞ ball semantically represents X. In case of a point cloud classifier, a point cloud x before and after applying any perturbation within the ℓ_∞-ball should look similar geometrically to a human eye (see Fig. 3). We use a similar notation as Madry et al. [13] where S is the set of all allowed perturbations in the ℓ_∞ ball. Now, we call a perturbed x as $x^{adv} = x + \delta$, where $\delta \in S$.

Within this threat model, AutoAttack performs three white-box attacks (APGD, Targeted-APGD, and FAB) and one black-box attack (Square attack). White-box attacks have full access to information of the model, while black-box attacks do not use any prior knowledge of a particular model. In our study, we utilize the APGD and Targeted-APGD(APGD-T) because square attack and FAB are image-specific attacks that could not be used for point clouds. In these white-box attacks, we begin with a clean, in-distribution data x. AutoAttack finds a $\delta \in S$ to find an x^{adv} that successfully fools the classifier. In the case of APGD, AutoAttack starts at a random point in the ℓ_∞-ball. It then performs iterations of projected gradient descent using a decaying learning rate α to find an x^{adv} that maximizes its loss function the most. It then performs several more random restarts in the same ℓ_∞-ball and repeats the same process to find the x_{adv} that has the highest loss. If the resulting x_{adv} fools the classifier correctly, the attack was successful for this particular data point x.

Beyond defining the threat model, AutoAttack is a parameter-free evaluation tool that allows us to easily measure the adversarial robustness of different models. Thus, we can use AutoAttack as our method to measure the adversarial robustness of point cloud classifiers.

3.2. Evaluating Adversarial Robustness

As suggested by Sun et. al [16], we use a threat model of ℓ_∞ norm with $\epsilon = 0.05$ to generate adversarial examples. Specifying this threat model, we can now use AutoAttack. In particular, we created a script that parses ModelNet40(non-corrupted) point cloud data into a list, load pre-trained models from the ModelNet40-C zoo, and run AutoAttack to generate the adversarial examples within

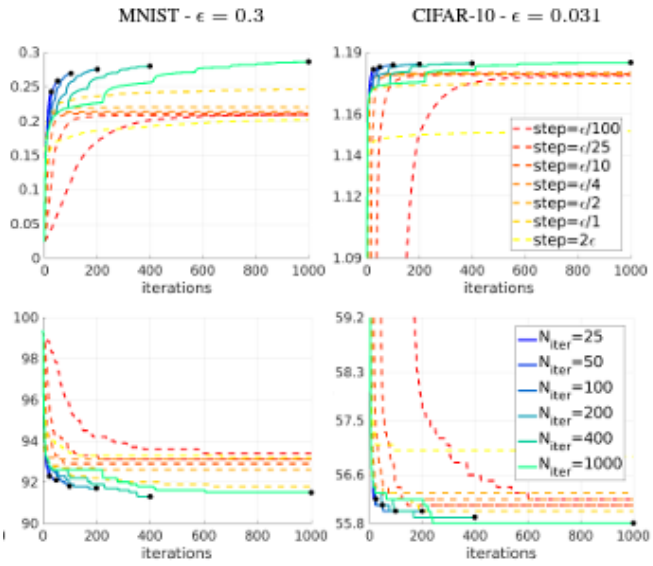

Figure 4. Best(highest) cross-entropy loss(top) and accuracy(bottom) obtained as a function of iterations for the TRADES model [19] for PGD with momentum(dotted lines) and APGD(solid lines) on MNIST and CIFAR dataset. [5]

the specified threat model. AutoAttack then runs the model with perturbed data to see if it successfully fools the classifier. Finally, AutoAttack reports how many of the input points were successfully perturbed to fool the classifier. We will collect this metric of adversarial robustness and compare it to their corruption robustness of various models that were already evaluated by the authors of ModelNet40-C. All in all, moost of our technical work in this section comprised mostly of creating a script to run AutoAttack tests and making sure that ModelNet40-C and AutoAttack frameworks were consistent with each other. Most notably, we implemented our own version of AutoAttack for point cloud classifiers, although AutoAttack was originally created for image classification.

3.3. APGD to Train Models

As an extension to our efforts of finding the relationship between adversarial and corruption robustness, we explored an alternative adversarial training method to evaluate its effect on the two robustness metrics. Specifically, we adversarially trained the point cloud classifiers using APGD.
The role of APGD in our training process is best explained through the objective function of adversarial training. As defined by Madry et al [13], the objective function is:

$$\min_\theta \rho(\theta), \text{ where } \rho(\theta) = \mathbb{E}_{(x,y)\sim D}[\max_{\delta \in S} L(\theta, x+\delta, y)]$$

This is a saddle optimization problem where the inner maximization problem finds the perturbation that produces the highest loss for each data point, while the outer minimization problem finds the weights and parameters of the network that minimize the expected adversarial loss. To solve the inner maximization problem, the traditional approach has been to use PGD. For point cloud classifiers, the adversarially trained model in ModelNet40-C has also employed PGD. However, using APGD in adversarial training has not been common, and we explore this in the domain of point cloud classifiers.

3.4. APGD vs PGD

Our motivation of using APGD in adversarial training is as follows. APGD is a relatively recent attack designed by the creators of AutoAttack [5] as an alternative to the traditional PGD attack. As mentioned earlier, both APGD and PGD search for a perturbation within an ℓ_∞-ball around an input point x to find an adversarial example that maximizes its loss function. Both attacks begin at a random point and apply projected gradient descent to a direction that maximizes a loss function. The key difference between APGD and PGD is that APGD automatically changes its step size α, which is the distance of travel within the ℓ_∞-ball between each iteration. For each step-size APGD keeps track of the perturbation δ_1 that generated the largest loss, and it restarts its maximization search at that point δ_1 when it decreases its step size. The phase when APGD has a big step size is called the exploratory phase that searches for a general maximum, while it gradually moves on to an exploitation phase with a smaller step size. Fig. 4 shows a concrete example of APGD outperforming standard PGD attacks in finding better adversarial examples. Croce and Hein [5] conclude that APGD is a better attack methodology in general. Thus, we believe APGD could improve the adversarial robustness of point cloud classifiers, while also improving the corruption classifiers.

3.5. Evaluation Metrics

For corruption robustness metric, we will use the Error Rate(ER), which is used in the ModelNet40-C paper. This is the percentage of corrupted or adversarial point clouds that were mis-classified by a classifier. Though accuracy is more conventional as the metric for adversarial robustness, we will also use the ER for adversarial robustness for consistency. Precisely, the ER is calculated by:

$$\text{ER} = 1 - \frac{\sum_{j=1}^{|D|} 1_{c_j == y_j}}{|D|}$$

where D is either a clean, corrupted, or adversarial dataset and c_j are the predicted labels of the jth input in D.

We note ER_{Adv} as the error rate for adversarial data, ER_{CR} as the error rate for corrupt data, and ER_{CL} as the error rate for clean data.

4. Experiments

In this section, we describe our evaluation on different models and provide an analysis. First, we examine the effectiveness of different data augmentation techniques for adversarial robustness and its relationship to corruption robustness. Moreover, we benchmark both corruption and adversarial robustness for different models including our model using adversarial training with APGD. Our result highlights the importance of both corruption and adversarial robustness metrics and encourages further research in this area, as our custom model shows potential for obtaining high robustness for all metrics.

4.1. Comparing Adversarial and Corruption Robustness

As described in 3.2, we ran our version of AutoAttack for point clouds against the models provided by the ModelNet40-C model zoo(DGCNN and PointNet). By doing so, we aimed to obtain a more clear vision of the relationship, if any, between the two robustness metrics. For each model, we obtain a ER_{adv}, which is the error rate for adversarial data. Then, we plotted ER_{adv} in comparison to the error rate against corruption(ER_{CR}) which was determined by Sun et al [17], as seen in Figure 5.

- *Insight 1: Corruption Robustness does not imply Adversarial Robustness*

From Figure 5, we can see that most models, despite their low error rate for corrupted data, have a very high error rate close to 100% for adversarial data. However, models that were adversarially trained using PGD showed relatively low ER_{adv} even though their ER_{CR} is comparable to other models with different data augmentation techniques. Thus, we can conclude their is no obvious relationship between adversarial and corruption robustness. Specifically, a high corruption robustness does not imply a high adversarial robustness.

Our initial intuition was that high corruption robustness would indicate adversarial robustness because of their semantic similarities. Particularly, some corruption types such as added noise were semantically similar to adversarially perturbed data. Thus, we speculated that adversarial data was a subset of corrupted data, and thus high corruption robustness would imply a lower adversarial robustness.

From our data, we found that our intuition was wrong possibly (1) because the threat model limits perturbed models to be much similar to the original models or (2) because the adversarial attacks deliberately target to increase the loss of the model to induce mis-classification, which arguably is much worse than simply corrupted data. This makes sense because, as stated by Croce et. al. in [5], adversarial training is one of the very few ways to create an adversarially

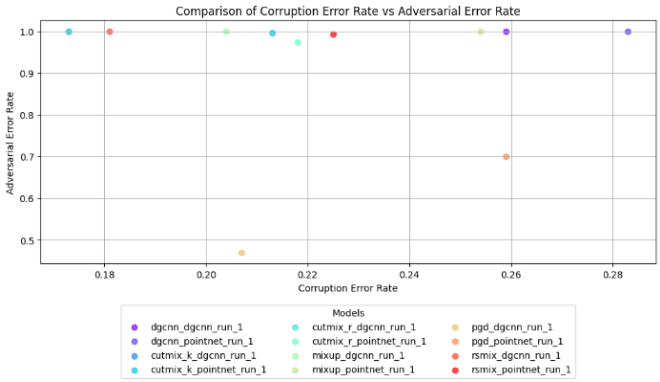

Figure 5. Comparison between Adversarial Error and Corruption Error Rate for DGCNN and Pointnet models using various data augmentation such as PGD, Cutmix-R etc.

robust model. Though the study by Croce et. al. was done under image classifications, it is consistent with our observations with point cloud classifiers - simply applying the various data augemntations such as PointCutMix-R and PointMixUp does not improve adversarial robustness even if it increases corruption robustness.

4.2. Adversarially Training a model using APGD

We used APGD for adversarial training models as an alternative to the traditional PGD, which was used by Sun et. al. [17] to improve both adversarial and corruption robustness. Between DGCNN and PointNet, we decided to apply APGD to DGCNN because DGCNN with PGD performed better in all robustness metrics against clean, adversarial, and corrupted data than PointNet with PGD.

Model Setup: In the interest of time and resources, we trained the model using APGD with an iteration number of 7 and for 50 epochs. The iteration number of APGD is the number of steps taken in the ℓ_∞-ball to generate an adversary. While APGD will find a better adversary with larger iterations, typically around 100, training the models will take much more time. However, since APGD will perform better with larger iterations, we expect the model with more APGD iterations to have higher adversarial robustness. There is a tradeoff between performance and time, and we chose 7 because it was the number of steps used for the pre-trained DGCNN model using PGD. Similarily, for the number of epochs, we reduced it by a lot because we lacked time and resources - typically, the pre-trained models from the ModelNet40-C are trained for 300 epochs.

Results: Figure 6 shows the validation and training accuracy of DGCNN models through epochs in our adversarial training using APGD. To clarify, both the validation and training dataset are from ModelNet40. For our main model with 7 APGD iterations and 50 epochs, we reach a validation accuracy of 89.3%. We also observe that the training

Figure 6. Training and Validation accuracy of DGCNN models using APGD with 15 iterations and 7 iterations over epochs. As a reference, we also have accuracies of a model trained with PGD for comparison. Training was cut short due to the lack of time and resources.

DataAug-ModelType	ER_{CL}	ER_{CR}	Density	Noise	Transformation	ER_{Adv}	Avg. ER
PointNet-PointCutMix-K	9	21.3	**26.8**	21.8	15.4	99.71	43.34
PointNet-PointCutMix-R	9.4	21.8	30.5	18	16.9	97.36	42.85
PointNet-PointMixup	8.9	25.4	28.3	28.9	19.0	100	44.77
PointNet-RSMix	9.8	22.5	24.8	27.3	15.5	99.32	43.87
PointNet-PGD	11.8	25.9	28.8	28.4	20.5	69.92	35.87
DGCNN-PointCutMix-K	**6.8**	**17.3**	29.1	11.9	10.9	100	41.37
DGCNN-PointCutMix-R	7.4	**17.3**	28.9	**11.4**	11.5	100	41.57
DGCNN-PointMixup	7.8	20.4	32.1	16.8	12.3	100	41.7
DGCNN-RSMix	7.1	18.1	28.8	13	12.6	99.9	125.1
DGCNN-PGD	8.1	20.7	36.8	13.8	11.5	46.97	25.26
DGCNN-APGD (ours)	11.92	20.03	33.68	16	**10.42**	**42.28**	**24.74**

Table 1. Error Rates for clean, corrupted, and adversarial data for DGCNN and PointNet models trained under various data augmentation techniques. Avg. ER is the average of the three main ER, which provides us a comprehensive view on its robustness.

accuracy is lower than the validation accuracy, which can be explained by how our model performs dropout during training.

Table 1 shows the performance of DGCNN and PointNet models with different data augmentation techniques. From left to right the table displays the model name, error rate(ER) for clean data, ER for corrupted data, ER for different subtypes of corrupted data (Density, Noise, Transformation), ER for adversarial attacks, and the average of the clean, corrupted, adversarial ER. Precisely,

$$\text{ER}_{\text{CR}} = \frac{1}{3}(\text{ER}_{\text{Density}} + \text{ER}_{\text{Noise}} + \text{ER}_{\text{Trans.}})$$

$$\text{Avg. ER} = \frac{1}{3}(\text{ER}_{\text{CL}} + \text{ER}_{\text{CR}} + \text{ER}_{\text{Adv.}})$$

Our results showed that our model achieved the lowest Avg. ER and ER_{adv}.

- *Insight 2: No one specific robustness metric gives a comprehensive view on the model performance.*

As shown by Table 1, the pre-trained models that are robust against corruption are not robust against adversarial data. Thus, claiming that these models are robust simply based on corruption error rate or clean error rate would be misleading. The same could be said about the other two robustness metrics. Thus, we believe it will be beneficial to use a metric that takes into account all types of robustness when training a model in order to obtain a more comprehensive insight into how well a model performs. In our table, we simply take an average of the error rates as our performance metric. However, we also believe a weighted average may be better in different use cases of the point cloud classifier so that the metric will emphasize the more important robustness metrics. Nonetheless, having a metric that captures all robustness is important because no individual metric can provide a full picture of the model performance.

- *Insight 3: APGD has potential for achieving a more robust model*

Table 1 demonstrates that our model adversarially trained with APGD had the lowest average error rate. Since our model was only trained for 50 epochs, we believe our method has the potential to improve even further if we train it for the standard 300 epochs. Furthermore, we believe another model that is adversarially trained using APGD could outperform our current model. For example, instead of using DGCNN, we believe that there is potential for APGD to perform really well in PCT (point cloud transformers), which has shown to perform well against corrupted datatypes [17].

Furthermore, from Figure 6, we can see that the loss values for DGCNN with APGD of iteration 15 is typically lower than the other two models. This can be explained how the APGD is able to generate a better adversarial example with more iterations. This may indicate that adversarial training with APGD with higher iteration number provides a better result in adversarial robustness in the longer run with more epochs. Thus, we believe it will be worthwhile investigating the effects of different iteration numbers on the overall error rates.

5. Discussion

In this section, we will go over open questions and future work based on our findings. Then, we will also discuss the key takeaways from our research.

5.1. Key takeaways

As a reminder, we offered the following insights:

- *Insight 1: Corruption Robustness does not imply Adversarial Robustness*
- *Insight 2: No one specific robustness metric gives a comprehensive view on the model performance*
- *Insight 3: APGD has potential for achieving a more robust model*

Key takeaway: When training a model (specifically for, but not limited to, point cloud classifiers), we encourage an explicit focus on improving both corruption and adversarial robustness since we cannot conclude that one guarantees the other. We specifically proposed an adversarial training method using APGD, but we encourage all further research that attempts to increase all robustness against clean, adversarial, and corrupted data.

5.2. Open Questions/Future work

Open Question 1: Does adversarial robustness imply corruption robustness? From insight 1, we concluded corruption robustness does not imply adversarial robustness. However, our observations may hint that the converse may be true - a high adversarial robustness may imply a high corruption robustness. However, we cannot conclude this, as we only have data for a few examples that support this ovservation. Thus, this is an area of future work. If we determine that adversarial robustness implies corruption robustness, we can focus solely on addressing adversarial robustness instead of conducting a dual optimization for both robustness metrics.

Open Question 2: What is the optimal iteration number for APGD in adversarial training point cloud classifiers? To train our model, we used an APGD with an iteration of 7 steps. When we increase the iteration numbers, we typically can generate adversaries with a higher loss value. Thus, if we apply APGD with higher iterations to adversarial training, we expect to see higher adversarial robustness. However, with higher adversarial robustness, there is typically a trade-off in the clean robustness metric, and potentially the corruption robustness. This would involve creating a heuristic for evaluating the different models based on all these robustness metrics such as Avg. ER, and testing many models with different APGD iteration numbers. Though we would like to test different models ourselves, we simply cannot test so many possible models since training a single model consumes a lot of time and resources. Thus, we encourage further research to find a way to determine an optimal iteration number for APGD in adversarial training point cloud classifiers.

Open Question 3: How can we modify advesarial training to improve corruption robustness? From our observations, adversarial training (with PGD and APGD) is the only method we tested that is robust to corruption and adversarial attacks. Furthermore, adversarial training is often referred as one of the best way to defend against adversarial attacks [5]. Then, naturally, we would want to use adversarial training to train our point cloud classifiers. However, adversarial training does poorly against certain corruption types, such as occlusion, that are not similar to adversarial examples. Thus, we encourage research on how to increase those areas of corruption robustness when applying adversarial training.

6. Conclusion

This study has taken the first steps toward understanding the relationship between corruption, adversarial, and clean robustness in point cloud classifiers. Our findings reveal that high corruption robustness does not necessarily translate to high adversarial robustness. This insight is crucial for the development of robust point cloud classifiers, especially in safety-critical applications like advanced driver assistance systems. Thus, we emphasize the need for a holistic view of model performance, taking into account various robustness metrics rather than relying on a single metric. Furthermore, we introduced the approach of using Auto-PGD (APGD) for adversarial training, which shows promising results in achieving a balance between all metrics of robustness. Future work in this domain should explore the potential of adversarial robustness as an indicator of corruption robustness, tune APGD parameters for training, and refine adversarial training methods to enhance specific areas of corruption robustness. Overall, this study encourages continued exploration and innovation in enhancing the robustness of point cloud classifiers, which is essential for their reliable deployment in real-world applications.

References

[1] Marius Preda Chao Cao. 3d point cloud compression: A survey. *The 24th International Conference on 3D Web Technology*, 2019. 1

[2] Hao Su Charles R. Qi, Li Yi and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems 30*, 2017. 1, 2, 3

[3] Kaichun Mo Charles R. Qi, Hao Su 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*, 2017. 1, 2, 3

[4] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees G. M. Snoek. Pointmixup: Augmentation for point clouds, 2020. 3

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *International conference on machine learning*, 2020. 3, 4, 5, 7

[6] Mingqiang Wei Kyle Gao Linlin Xu Dening Lu, Qian Xie and Jonathan Li. Transformers in 3d point clouds: A survey. *arXiv preprint arXiv:2205.07417*, 2022. 1, 2

[7] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline, 2021. 3

[8] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Apr. 2021. 3

[9] Houwen Peng Jinjie Mai Hasan Abed Al Kader Hammoud Mohamed Elhoseiny Guocheng Qian, Yuchen Li and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems 35*, 2022. 1, 2

[10] Ivan V. Bajić Hanieh Naderi. Adversarial attacks and defenses on 3d point cloud classification: A survey. *arXiv:2307.00309*, 2023. 3

[11] Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee. Regularization strategy for point cloud via rigidly mixed sample, 2021. 3

[12] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis, 2019. 3

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. 2, 3, 4

[14] Felix Juefei-Xu Qing Guo Xingyu Li Lei Ma Shuangzhi Li, Zhijie Wang. Common corruption robustness of point cloud detectors: Benchmark and enhancement. *arXiv preprint arXiv:2210.05896*, 2022. 2

[15] Zachary Patterson Sneha Paul and Nizar Bouguila. Dualmlp: a two-stream fusion model for 3d point cloud classification. *Vis Comput*, 2023. 2

[16] Jiachen Sun, Yulong Cao, Christopher B Choy, Zhiding Yu, Anima Anandkumar, Zhuoqing Morley Mao, and Chaowei Xiao. Adversarially robust 3d point cloud recognition using self-supervisions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15498–15512. Curran Associates, Inc., 2021. 3

[17] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z. Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions, 2022. 1, 2, 3, 5, 6

[18] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration, 2019. 3

[19] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy, 2019. 4

[20] Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujing Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification, 2021. 3

[21] Aditya Khosla-Fisher Yu Linguang Zhang Xiaoou Tang Jianxiong Xiao Zhirong Wu, Shuran Song. 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1