Improving Robustness of 3D Point Cloud Recognition from a Fourier Perspective

Yibo Miao[1,2], **Yinpeng Dong**[3,6†], **Jinlai Zhang**[4], **Lijia Yu**[5], **Xiao Yang**[3], **Xiao-Shan Gao**[1,2†]
[1] KLMM, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Tsinghua University, Beijing 100084, China
[4] Changsha University of Science and Technology, Changsha 410114, China
[5] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China [6] RealAI
miaoyibo@amss.ac.cn, dongyinpeng@tsinghua.edu.cn, xgao@mmrc.iss.ac.cn

Abstract

Although 3D point cloud recognition has achieved substantial progress on standard benchmarks, the typical models are vulnerable to point cloud corruptions, leading to security threats in real-world applications. To improve the corruption robustness, various data augmentation methods have been studied, but they are mainly limited to the spatial domain. As the point cloud has low information density and significant spatial redundancy, it is challenging to analyze the effects of corruptions. In this paper, we focus on the frequency domain to observe the underlying structure of point clouds and their corruptions. Through graph Fourier transform (GFT), we observe a correlation between the corruption robustness of point cloud recognition models and their sensitivity to different frequency bands, which is measured by the GFT spectrum of the model's Jacobian matrix. To reduce the sensitivity and improve the corruption robustness, we propose Frequency Adversarial Training (FAT) that adopts frequency-domain adversarial examples as data augmentation to train robust point cloud recognition models against corruptions. Theoretically, we provide a guarantee of FAT on its out-of-distribution generalization performance. Empirically, we conducted extensive experiments with various network architectures to validate the effectiveness of FAT, which achieves the new state-of-the-art results.

1 Introduction

3D point cloud recognition based on deep neural networks (DNNs) [35, 36, 65] has achieved unprecedented performance on typical benchmarks [5, 67], which assume that the data are independently and identically distributed. However, in practical scenarios, point clouds suffer from severe corruptions (e.g., noise, density change, transformation) due to sensor imprecision and scene complexity [66, 76]. When the testing distribution is different from the training distribution caused by corruption, point cloud recognition models have significant performance degradation [41, 51], indicating that they lack the robustness of human visual system [40], while also raising concerns about safety and reliability of these models. As deep 3D point cloud recognition has been increasingly deployed in safety-critical applications, such as autonomous driving [6, 84], robotics [60, 92], and medical image processing [54], it is of crucial importance to improve the robustness of 3D point cloud recognition models to out-of-distribution (OOD) point cloud data induced by corruptions [10].

†Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-2177

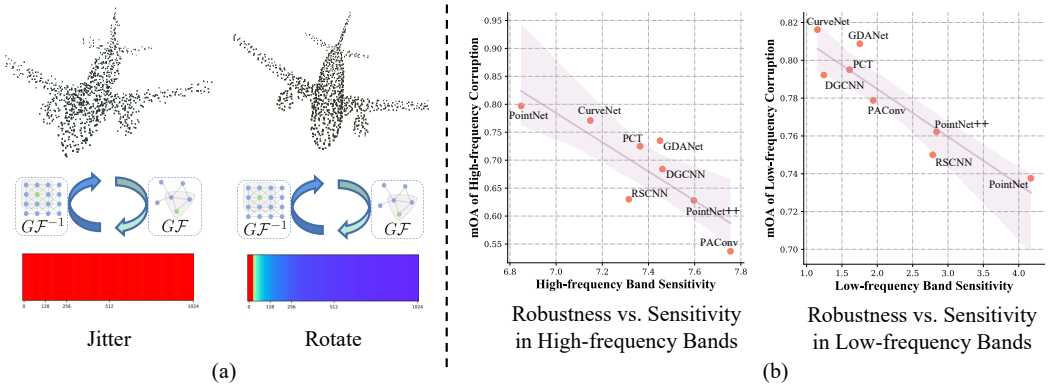


Figure 1: **(a):** The graph frequency-domain representations of "Jitter" and "Rotate" in ModelNet-C [41]. "Jitter" has higher power in the high-frequency region, while "Rotate" has higher power in the low-frequency region. **(b):** The relationship between the corruption robustness (measured by mean overall accuracy (mOA) [41]) of various models and the sensitivity to high/low frequency bands. Our proposed high/low frequency sensitivity metric is negatively correlated with the model's robustness under high/low frequency corruptions.

To improve the corruption robustness, the most effective approaches to date are based on carefully designed data augmentation techniques [41, 51]. Inspired by 2D image augmentations [85, 89], some methods blend two point clouds for data augmentation using shortest-path interpolation (e.g., PointMixup [7]), random blending (e.g., PointCutMix-R [90]), and rigid transformation (e.g., RSMix [24], PointCutMix-K [90]). PointWOLF [22] enriches data diversity by applying non-rigid deformation to object parts. WOLFMix [41] deforms objects first and then rigidly blends two deformed objects. Although these data augmentation techniques improve the corruption robustness to some extent, they are all based on spatial-domain transformations. Raw point clouds in the spatial domain have low information density and heavy spatial redundancy [8], making it challenging to analyze which specific information is corrupted. To address this challenge, we shift our attention from the spatial domain to the frequency domain to analyze the underlying structure of point clouds that is not easily observable from the raw point clouds. In the frequency domain, point clouds are compactly represented, facilitating a better understanding of low-level distortions that are free of high-level semantics.

To design robust models, the first step is to understand how corruption is represented in the frequency domain. We achieve this by transforming the raw point clouds and the corresponding corruptions into compact representations in the frequency domain using the graph Fourier transform (GFT) [43]. By visualizing the transformed signals, we observe that different corruptions affect varying frequency bands, as shown in Fig. 1(a). Motivated by the differences, we investigate the relationship between the corruption robustness of various point cloud recognition models and their sensitivity to different frequency bands [81]. To measure the sensitivity, we design **a novel metric based on GFT spectrum of the Jacobian matrix** of the model, as shown in Fig. 3. Our key insight is that **our proposed high/low frequency sensitivity metric is negatively correlated with the model's robustness under high/low frequency corruptions**, as shown in Fig. 1(b). This correlation emphasizes the importance of the model's sensitivity to high and low frequencies for corruption robustness. However, it is still challenging to simultaneously reduce the sensitivity of point cloud recognition models to both high and low frequencies.

To address this issue, we propose **Frequency Adversarial Training (FAT)** to improve the corruption robustness of 3D point cloud recognition models. FAT trains a model with adversarial examples that add perturbations to the frequency-domain representations of point clouds. Intuitively, a model robust to worst-case perturbations should be more resistant to real-world corruptions [72, 80]. We provide **a theoretical analysis that demonstrates the effectiveness of FAT in ensuring OOD generalization of the model**, as shown in Theorem 1. To eliminate potential performance degradation due to mutual interference between high and low frequency signals, we utilize the AdvProp training framework [72], based on which we use three separate batch normalization (BN) statistics for clean samples, high-frequency adversarial samples, and low-frequency adversarial samples, respectively.

We conducted extensive experiments to validate the effectiveness of our approach in improving the robustness of point cloud recognition models under common corruptions [41, 51]. With various

network architectures, our method improves the corruption robustness by a large margin. By integrating our approach with previous data augmentation techniques, we achieve the new state-of-the-art performance.

2 Related work

Deep learning on 3D point clouds. Deep 3D point cloud recognition [16, 35, 38, 55, 70, 73, 79] has emerged in recent years as a prominent research area with diverse applications in several fields such as 3D object classification [46, 83, 86], 3D scene segmentation [20, 64, 75], and 3D object detection in autonomous driving [77, 95]. One of the pioneering works is PointNet [35], which employs a multilayer perceptron to learn point features and utilizes a max-pool module to aggregate them efficiently. Many subsequent works [13, 30, 36, 78] improve upon PointNet. Several approaches focus on designing special convolutions on 3D domains [26, 31, 56] or developing graph neural networks [14, 44, 65] to improve point cloud recognition, such as DGCNN [65] which builds a dynamic graph for point cloud data. Recently, drawing inspiration from research in the frequency domain [4, 49, 61, 81], GDANet [74] introduces a geometry-disentangle module to dynamically separate point clouds into the contour and flat parts of 3D objects, thereby capturing complementary 3D geometric semantics. PCT [17] uses Transformer to improve point cloud learning. Additionally, there is a growing discussion on point cloud augmentation, including mix-based augmentations [7, 90], deformation-based augmentations [22], and auto-augmentations [25].

Robustness in 3D point cloud recognition. Following the previous studies on robustness in the 2D image domain [53, 3, 15, 19, 32, 42, 59, 9, 82], several works [18, 50, 52, 63, 69, 34, 97] have explored the robustness of 3D point cloud classifiers. Concerning adversarial robustness, Xiang et al. [69] first demonstrate that point cloud recognition is vulnerable to adversarial point generation attacks. Further research [21, 28, 29, 57, 91, 2] has employed gradient-based point perturbation attacks. Some defensive techniques are proposed, such as input randomization [12, 93] and geometry-aware framework [68] to defend against such vulnerabilities. Sun et al. [47, 48] have studied the effectiveness of adversarial training and pre-training on self-supervised tasks in enhancing robustness. In terms of corruption robustness, some works have studied the problem using invariant feature extraction [71], and adaptive sampling [76]. Recently, two benchmarks [41, 51] are developed for the robustness of 3D point cloud recognition under corruptions and demonstrate the effectiveness of data augmentation. However, unlike the existing spatial-domain data augmentation techniques [22, 25], in this paper, we focus on the frequency domain and propose Frequency Adversarial Training (FAT) to improve the model's out-of-distribution generalization ability.

3 Methodology

The existing 3D point cloud recognition models exhibit significant performance degradation under point cloud corruptions [41, 51]. Although data augmentation techniques have shown the effectiveness in improving robustness, they are typically based on spatial-domain transformations, which suffer from low information density and heavy spatial redundancy of the raw point clouds. Consequently, it is difficult to analyze which specific information has been lost due to corruptions within the spatial domain. To address this challenge, we shift our focus to the frequency domain, which enables us to analyze the underlying structure of point clouds.

In the following, we first provide the background knowledge of graph Fourier transform (GFT) in Sec. 3.1, then analyze the point cloud corruptions in the frequency domain in Sec. 3.2, and investigate the relationship between the model's corruption robustness and sensitivity to frequency changes in Sec. 3.3. Based on the analyses, we propose a Frequency Adversarial Training (FAT) method detailed in Sec. 3.4 with a theoretical analysis to guarantee its effectiveness in Sec. 3.5.

3.1 Graph Fourier transform

Images are typically transformed and recovered in the frequency domain with the 2D discrete Fourier transform (DFT) and inverse DFT [39]. Unlike images, although 3D point clouds are highly structured, they reside on irregular domains without an ordering of points, hindering the deployment of traditional Fourier transforms. However, graphs provide a natural and accurate representation of

https://doi.org/10.52202/079017-2177

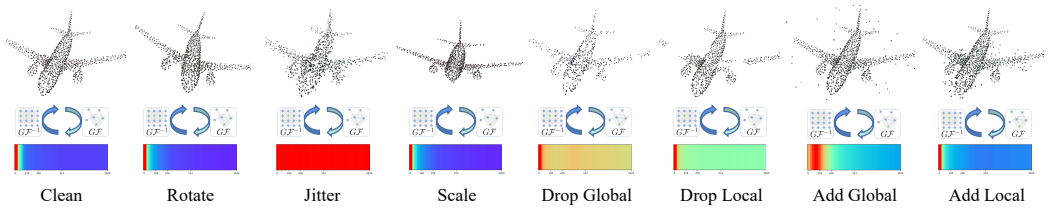


Figure 2: The leftmost image displays the graph frequency-domain representation of the raw point clouds. To estimate the expected value of $\mathbb{E}_{\mathcal{P}}[|G\mathcal{F}(\mathcal{P})|]$, we average over all validation point clouds in ModelNet40 [67]. The frequencies are arranged from left to right in ascending order. The other seven images display the graph frequency-domain representations of each corruption in ModelNet-C [41]. The raw point clouds exhibit higher power in the low-frequency region. The corruption "Jitter" has much higher power in the high-frequency region. The power of corruptions such as "Rotate" and "Scale" is concentrated on the low-frequency components.

irregular point clouds. Once a graph is constructed to represent the point cloud, the graph Fourier transform (GFT) [43] can compactly transform it into the frequency domain.

Given a point cloud $\mathcal{P} := \{\boldsymbol{p}_i\}_{i=1}^{n} \in \mathbb{R}^{n\times 3}$ of n points, where $\boldsymbol{p}_i$ denotes the xyz coordinates of a point, we construct a directed graph $\mathcal{G} = \{\mathcal{P}, \mathcal{E}, \boldsymbol{W}\}$ to represent it. The graph consists of a vertex set $\mathcal{P}$, an edge set $\mathcal{E}$ connecting the vertices, and an adjacency matrix $\boldsymbol{W}$. The entry $w_{i,j}$ in the adjacency matrix represents the weight of the edge from vertices i to j, which is used to capture the similarity between adjacent vertices. Here, we construct a weighted k-nearest neighbor graph (i.e., each vertex is only connected to its k-nearest neighbors) using the Euclidean distance $d_{ij} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2$ between vertices i and j, and the weight of the edge is $w_{i,j} = e^{-d_{ij}^2}$.

After constructing the graph representation of the point cloud, we focus on the combinatorial graph Laplacian [45], defined as $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix with the i-th diagonal entry $d_{i,i} = \sum_{j=1}^{n} w_{i,j}$ representing the degree of the i-th node. $\boldsymbol{L}$ is symmetric and positive semi-definite, and can be eigen-decomposed as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, where $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_n]$ is an orthogonal matrix containing the eigenvectors $\boldsymbol{u}_i$, and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_n)$ is a diagonal matrix containing the eigenvalues. The eigenvalues are sorted in ascending order, representing frequencies from low to high. For a point cloud $\mathcal{P}$, the graph Fourier transform (GFT) can be applied to transform it into a compact representation in the frequency domain: $\hat{\mathcal{P}} = G\mathcal{F}(\mathcal{P}) := \boldsymbol{U}^\top\mathcal{P}$. The low-frequency components represent the coarse shape of the point cloud, while the high-frequency components represent the fine details. The inverse graph Fourier transform (IGFT) can be used to recover the point cloud in the spatial domain as $\mathcal{P} = G\mathcal{F}^{-1}(\hat{\mathcal{P}}) := \boldsymbol{U}\hat{\mathcal{P}}$.

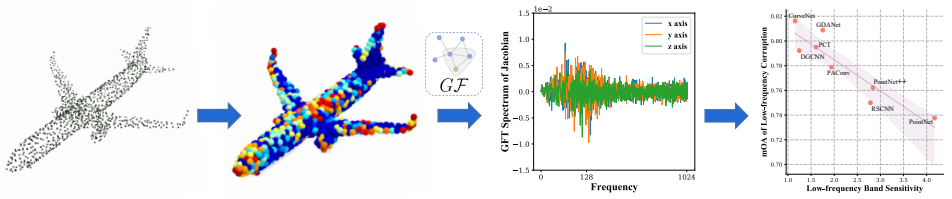
3.2 Analyzing point cloud corruptions in the frequency domain

We employ GFT to transform point clouds and their corruptions into compact representations in the frequency domain, allowing us to analyze the underlying structures of these low-level distortions that are hardly observable in the spatial domain. For raw point clouds, we transform them to the frequency-domain representations and calculate $\mathbb{E}_{\mathcal{P}}[|G\mathcal{F}(\mathcal{P})|]$ by averaging over all validation point clouds in ModelNet40 [67]. For each corruption type in ModelNet-C [41], we calculate $\mathbb{E}_{\mathcal{P}}[|G\mathcal{F}(\mathcal{C}(\mathcal{P}) - \mathcal{P})|]$ similarly, where $\mathcal{C}$ denotes the corruption function. As the input point clouds have three spatial axes (x, y, z), we take the average over these channels. In Fig. 2, we visualize the graph frequency-domain representations of raw point clouds and the corruptions in ModelNet-C. We can see that the raw point clouds have higher power in the low-frequency region, while the corruption "Jitter" leads to higher power in the high-frequency region. For corruptions such as "Rotate" and "Scale", the corrupted power is concentrated more on the low-frequency components. The results demonstrate that different corruptions of point clouds affect different frequency bands.

3.3 Relationship between corruption robustness and sensitivity to frequency bands

Motivated by the different effects of corruptions on varying frequency bands observed in the graph frequency-domain representations, we investigate the relationship between the corruption robustness of 3D point cloud recognition models and their sensitivity to different frequency bands.

https://doi.org/10.52202/079017-2177



Input | Jacobian Matrix | Jacobian Matrix in GFT Spectrum | Robustness vs. Sensitivity in Low-frequency Bands

Figure 3: An illustration of computing the Fourier spectrum of the Jacobian matrix for a single input point cloud. First, the Jacobian matrix for the input point cloud is computed. The gradient value of the output loss is visualized for each point. A higher gradient value (skewed to red) indicates that the model is more sensitive to changes at that point. Next, we utilize Graph Fourier Transform (GFT) on the Jacobian matrix to obtain a compact representation and measure its sensitivity in the Fourier domain. Finally, by examining the sensitivity measurement of different point cloud models in different frequency bands, we construct a relationship diagram with natural robustness.

To measure the sensitivity of a model on different frequency bands, we propose to perform graph Fourier transform (GFT) on the Jacobian matrix of the model's output loss with respect to its input point cloud. Intuitively, the Jacobian matrix represents how the model's output changes with small variations in its input point cloud, revealing its sensitivity to different points in the spatial domain [1]. With GFT, we can obtain the frequency-domain representation of the Jacobian matrix, which reveals the model's sensitivity to different frequency bands of input. If a model's Jacobian matrix has a high proportion of low/high frequency components, it will be sensitive to low/high frequency bands.

Fig. 3 illustrates the computation of the frequency-domain Jacobian matrix for a single point cloud. Specifically, given an input point cloud $\mathcal{P}$, a classification model h, and a standard cross-entropy loss function $\mathcal{L}_h$ for the classification task, the Jacobian matrix $\mathcal{J}(\mathcal{P}) := \nabla_{\mathcal{P}}\mathcal{L}_h$ of the loss with respect to the input point cloud can be calculated. We then perform GFT on $\mathcal{J}(\mathcal{P})$ to obtain its frequency-domain representation, denoted as $\widehat{\mathcal{J}(\mathcal{P})} = \boldsymbol{U}^\top \mathcal{J}(\mathcal{P})$ in a compact form, using the original point cloud's neighborhood relations and feature vector matrix. Since the input point cloud has three axis channels (x, y, z), we take the average of these channels and normalize the result. We measure the model's sensitivity to input perturbations in the low-frequency band by summing the squares of the amplitudes of the first λ frequencies of the Jacobian matrix's graph Fourier spectrum. The sensitivity to high-frequency perturbations is measured by summing the squares of the amplitudes of the remaining $1024 - \lambda$ frequencies. A higher value of the metric indicates greater sensitivity to perturbations in that frequency band.

We can now measure the importance of the sensitivity to different frequency bands of point cloud recognition models on their corruption robustness. First, we measure and establish the relationship between sensitivity to high/low frequency bands of different point cloud models and their accuracy under high/low frequency corruptions. As illustrated in Fig. 1(b), our proposed frequency sensitivity metrics are negatively correlated with the corruption robustness. Therefore, point cloud models that are less sensitive to high/low frequency bands exhibit better robustness to high/low frequency corruptions. This correlation indicates that the sensitivity of models to different frequency bands affects their corruption robustness, providing insights for further improving the robustness of point cloud recognition models.

3.4 Frequency adversarial training

The above analyses highlight the importance of the sensitivity of point cloud recognition models to high and low frequencies on their corruption robustness. However, reducing the sensitivity of point cloud models to both high and low frequencies is still challenging. To address this problem, we propose **Frequency Adversarial Training (FAT)** to improve the corruption robustness of point cloud recognition models using adversarial examples in the frequency domain. Intuitively, a model trained to be robust to worst-case adversarial perturbations should be naturally robust to real-world corruptions [72, 80], as also theoretically demonstrated in Sec. 3.5.

To simultaneously reduce the sensitivity of point cloud recognition models to high and low frequencies, we generate high-frequency adversarial examples and low-frequency adversarial examples, which are added to the training set. We generate high-frequency adversarial examples that alter the details of

https://doi.org/10.52202/079017-2177

the point clouds, and low-frequency adversarial examples that change the rough shapes of the point clouds. To prevent the mutual interference of high-frequency and low-frequency adversarial examples that may lead to a decrease in model performance, we adopt the AdvProp training framework [72], where clean samples, high-frequency adversarial samples, and low-frequency adversarial samples are separately processed using three batch normalizations during adversarial training. Specifically, for an input point cloud $\mathcal{P}$ with the ground-truth label y, our optimization objective is

$$\arg\min_{\theta} \Big[\mathbb{E}_{(\mathcal{P},y)\sim\mathbb{D}}\Big(\mathcal{L}_h(\theta,\mathcal{P},y) + \max_{\epsilon_h\in\mathbb{S}_h} \mathcal{L}_h(\theta, G\mathcal{F}^{-1}(G\mathcal{F}(\mathcal{P})+\epsilon_h),y) + \max_{\epsilon_l\in\mathbb{S}_l} \mathcal{L}_h(\theta, G\mathcal{F}^{-1}(G\mathcal{F}(\mathcal{P})+\epsilon_l),y)\Big)\Big], \tag{1}$$

where $\mathbb{D}$ is the underlying data distribution, $\mathcal{L}_h$ is the loss function, θ is the network parameter, ϵ_h and ϵ_l are high-frequency and low-frequency adversarial perturbations, and $\mathbb{S}_h$ and $\mathbb{S}_l$ are the high-frequency and low-frequency perturbation ranges, respectively.

3.5 Theoretical analysis

To verify the claim that a model robust to frequency-domain worst-case perturbations should be more resistant to real-world corruptions, we provide a theoretical analysis that demonstrates the effectiveness of FAT in ensuring OOD generalization of the model.

Suppose (x, y) is a pair of training sample x and its label y. The loss on (x, y) with model parameter θ is $\mathcal{L}(\theta, (x, y))$, where $\mathcal{L}(\theta, (x, y))$ is continuous and differentiable for both θ and (x, y). We let $f(x) := \mathcal{L}(\theta, (x, y))$ for simplicity. Let $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and inverse Fourier transform, respectively. The norm $\|\cdot\|_p$ denotes the ℓ_p-norm. We have the following theorem:

Theorem 1. *If f satisfies that: $f(x) \in [0, M]$ for all x, $|f(\mathcal{F}^{-1}(\mathcal{F}(x)+\alpha)) - f(x)| \le \epsilon$ for all x and $\|\alpha\|_p \le \delta$, then for any distribution P_o and P_a satisfying that $Was^p(P_o, P_a) := (\inf_{u\in\Pi(P_o,P_a)} \mathbb{E}_{(x,z)\sim u}[\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p])^{1/p} \le \eta$, where $\eta < \delta$, then, with probability $1-\gamma$, we have:*

$$\mathbb{E}_{z\sim P_o}[f(z)] - \frac{1}{m}\sum_{i=1}^{m} f(x_i) \le \epsilon\left(1 - \frac{\eta^p}{\delta^p} - \sqrt{\frac{\ln(4/\gamma)}{2m}}\right) + \frac{\eta^p}{\delta^p}M + 4M\sqrt{\frac{\ln(4/\gamma)}{2m}}, \tag{2}$$

where $\{x_i\}_{i=1}^m$ are i.i.d. samples from P_a.

Remark 1. *Intuitively, OOD corresponds to the shifted distribution P_o that approaches the training distribution P_a. Thus $Was^p(P_o, P_a)$ defines OOD from the perspective of measuring the distance between distributions. $\mathbb{E}_{z\sim P_o}[f(z)] - \frac{1}{m}\sum_{i=1}^{m} f(x_i)$ represents the OOD generalization error of the model. $|f(\mathcal{F}^{-1}(\mathcal{F}(x)+\alpha)) - f(x)| \le \epsilon$ and $\|\alpha\|_p \le \delta$ indicate that the model is robust under frequency-domain perturbations. The bound (2) implies that models that are adversarially robust in the frequency-domain have smaller generalization bounds on OOD data.*

The proof of Theorem 1 is deferred to Appendix A. Thus, the frequency-domain adversarial robustness of the model guarantees the generalization on OOD data. We have the following observations:

- The right-hand side of Eq. (2) implies that models that are more robust to frequency domain adversarial samples (i.e., larger δ and smaller ϵ) have smaller OOD generalization bounds and thus perform better on OOD data.
- For Eq. (2), a larger number of training samples m leads to a smaller OOD generalization bound. This indicates that more training samples can compensate for the degradation of generalization performance.

4 Experiments

In this section, we first detail the experimental settings in Sec. 4.1, then present the main results in Sec. 4.2 to show the effectiveness of our method. We further integrate our method with other data augmentation techniques in Sec. 4.3 and perform ablation studies in Sec. 4.4.

Table 1: Quantitative results of vanilla training, adversarial training, DUP Defense and our proposed Frequency Adversarial Training (FAT) on the ModelNet-C test set. Our proposed FAT outperforms all other methods in terms of mean corruption error (mCE), which demonstrates the effectiveness of FAT for improving corruption robustness.

	Method	OA ↑	**mCE** ↓	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
DGCNN	Vanilla Training	0.926	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Adv Training	0.925	0.926	1.019	0.582	1.043	0.996	1.101	0.871	0.869
	DUP Defense	0.906	0.905	1.112	0.902	1.181	1.048	1.483	0.285	0.327
	FAT (Ours)	0.925	**0.825**	0.898	0.453	0.989	0.931	0.971	0.773	0.760
PointNet	Vanilla Training	0.907	1.422	1.902	0.642	1.266	0.500	1.072	2.980	1.593
	Adv Training	0.904	1.372	1.851	0.563	1.287	0.448	1.077	2.888	1.487
	DUP Defense	0.876	1.246	2.088	0.668	1.649	0.802	1.396	0.966	1.153
	FAT (Ours)	0.902	**1.237**	1.553	0.370	1.606	0.448	1.097	2.583	1.004
PCT	Vanilla Training	0.930	0.925	1.042	0.870	0.872	0.528	1.000	0.780	1.385
	Adv Training	0.919	0.976	1.042	0.389	1.074	0.911	1.193	1.108	1.116
	DUP Defense	0.919	0.925	1.112	0.699	1.043	0.738	1.261	0.410	1.215
	FAT (Ours)	0.920	**0.907**	1.009	0.345	1.085	0.843	1.237	0.912	0.920
GDANet	Vanilla Training	0.934	0.892	0.981	0.839	0.830	0.794	0.894	0.871	1.036
	Adv Training	0.926	0.960	1.112	0.506	1.032	0.927	1.140	1.064	0.938
	DUP Defense	0.915	0.897	1.140	0.788	1.064	0.698	1.179	0.427	0.985
	FAT (Ours)	0.928	**0.850**	1.167	0.408	0.926	0.794	1.111	0.654	0.887

4.1 Experimental setup

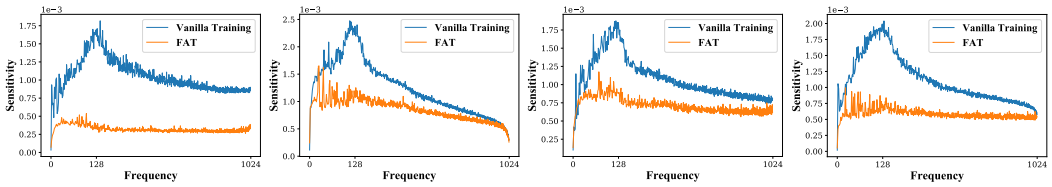
Dataset. To validate the effectiveness of our FAT method in enhancing the corruption robustness of 3D point cloud recognition models, we train all models on the standard ModelNet40 training set [67]. In addition to reporting the performance of the models on the original ModelNet40 validation set, we also evaluate the corruption robustness on ModelNet-C [41] in the main paper and ModelNet40-C [51] in Appendix B. The ModelNet40 dataset [67] contains 12,311 CAD models with 40 common object categories in the real world. We use the official split [35], where 9,843 examples are used for training and the remaining 2,468 examples are used for testing. The ModelNet-C dataset [41] is designed for measuring the network robustness to common point cloud corruptions. It consists of 7 different corruption types, including "Scale", "Jitter", "Rotate", "Drop Global", "Drop Local", "Add Global", and "Add Local". Each type of corruption has five severity levels. ModelNet40-C [51] is a similar dataset with 15 corruptions, which will be detailed in Appendix B.

Model architectures. Following [41, 51], we select four representative model architectures: PointNet [35], DGCNN [65], PCT [17], and GDANet [74]. These models represent different architectural designs and have been widely applied in 3D visual tasks.

Evaluation metrics. To measure the corruption robustness of different methods, we follow [41] and use the mean corruption error (mCE) as the main evaluation metric. We adopt the official baseline DGCNN and first compute the corruption error (CE) for a given corruption type i by averaging over 5 severity levels: $\mathrm{CE}_i = \frac{\sum_{l=1}^{5}(1-\mathrm{OA}_{i,l})}{\sum_{l=1}^{5}(1-\mathrm{OA}_{i,l}^{\mathrm{DGCNN}})}$, where $\mathrm{OA}_{i,l}$ is the overall accuracy on a corruption test set i at severity level l, and $\mathrm{OA}_{i,l}^{\mathrm{DGCNN}}$ is the overall accuracy of the baseline. Then, we average over the 7 corruption types to compute the mean corruption error: $\mathrm{mCE} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{CE}_i$. In addition, we also report the clean overall accuracy (OA), the corruption overall accuracy (mOA), and the relative mCE (RmCE) following [41]. Due to space constraints, we provide the definition of RmCE and report mOA and RmCE in Appendix B.

Implementation details. For each method, we train 250 epochs using the smooth cross-entropy loss [65] and Adam optimizer [23], and select the best performant model for further evaluation. We follow the DGCNN protocol [16]. For our method, we set $k = 30$ for the k-nearest neighbor graph and $\lambda = 100$ for dividing high-frequency and low-frequency [29]. We use PGD [33] and AOF [27] to generate high-frequency adversarial examples and low-frequency adversarial examples, respectively. We constrain $\mathbb{S}_h$ and $\mathbb{S}_l$ by 0.3 and 0.5, respectively. For more detailed training settings, please refer to Appendix B.

https://doi.org/10.52202/079017-2177



(a) DGCNN (b) PointNet (c) PCT (d) GDANet

Figure 4: Visualization of the sensitivity maps based on Jacobian matrices of Frequency Adversarial Training (FAT) and vanilla training under four different model architectures. FAT reduces the model sensitivity to different frequency bands, thereby enhancing their robustness to corruptions.

4.2 Main results

In this section, following [41, 51], we compare our proposed Frequency Adversarial Training (FAT) method with vanilla training, adversarial training and DUP Defense [93] on the ModelNet-C test set, demonstrating the effectiveness of FAT in enhancing corruption robustness. Table 1 presents a comparative analysis of different methods based on mean corruption error (mCE), clean overall accuracy (OA), and corruption error (CE) for each corruption type.

As shown in Table 1, our proposed Frequency Adversarial Training (FAT) outperforms all other methods in terms of mean corruption error (mCE), while exhibiting comparable performance in terms of overall accuracy (OA). The improvement in corruption robustness across the four different model architectures demonstrates the generalizability/universality of our method across different architectures. In Fig. 4, we visualize the sensitivity maps based on Jacobian matrices of Frequency Adversarial Training (FAT) and vanilla training under four different model architectures. FAT reduces the sensitivity of the model across different frequency bands.

It is noteworthy that GDANet introduces a geometry-disentangle module to dynamically disentangle point clouds into the contour and flat part of 3D objects, capturing complementary 3D geometric semantics. In contrast, FAT does not modify the network architecture to focus on the frequency domain but instead employs adversarial training in the frequency domain. As shown in Table 1, the two methods are complementary and synergistic, leading to improved model robustness. We report the performance of different methods in terms of overall corruption accuracy (mOA) and relative mCE (RmCE) in Appendix B, where the improvement in robustness of FAT is also significant under these metrics. The comparisons in Table 1 and Appendix B confirm that our proposed FAT enhances the OOD generalization ability of the model.

4.3 Data augmentation

To further validate the effectiveness of our proposed Frequency Adversarial Training (FAT), following [41], we investigate the performance of FAT in combination with different data augmentation strategies, including RSMix [24], PointWOLF [22], and WOLFMix [41]. These strategies respectively represent mix-based augmentation, deformation-based augmentation, and a combination of both mix-based and deformation-based augmentation. RSMix involves rigidly blending two point clouds using a transformation. PointWOLF enriches data diversity by applying non-rigid deformations to object parts. WOLFMix, designed based on PointWOLF and RSMix, first deforms the objects and then rigidly blends two deformed objects. When combining the data augmentation strategies, we first perform data augmentation on the input point cloud and then generate adversarial examples. For mix-based augmentation, we perform untargeted adversarial attacks on both labels being mixed to generate the adversarial examples.

In Table 2, we show the performance of FAT when integrated with different data augmentation strategies in terms of mean corruption error (mCE), clean overall accuracy (OA), and corruption error (CE) for each corruption type. Compared with a single data augmentation strategy, the combination of FAT and data augmentation strategies achieves a better mCE, which is attributed to the complementary and compatible information from both the spatial and frequency domains. The improvement in corruption robustness under three different data augmentation strategies demonstrates the generalization capability of our proposed method. As shown in Table 2, training GDANet with the combination of our proposed FAT with WOLFMix achieves a new state-of-the-art performance, with an impressive 0.537 mCE.

Table 2: Quantitative results of combining FAT with different data augmentation strategies on the ModelNet-C test set. Compared with a single data augmentation strategy, the combination of FAT and different data augmentation strategies achieves a better mCE. Training GDANet with the combination of our proposed FAT with WOLFMix achieves the new state-of-the-art performance, with an impressive **0.537 mCE**.

	Method	OA ↑	**mCE** ↓	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
PointNet	Vanilla Training	0.907	1.422	1.902	0.642	1.266	0.500	1.072	2.980	1.593
	RSMix	0.902	1.276	1.372	0.532	2.234	0.593	1.145	2.241	0.815
	RSMix+FAT (Ours)	0.904	**1.084**	1.340	0.389	1.670	0.415	0.899	2.241	0.636
	PointWOLF	0.902	1.311	0.912	0.633	2.128	0.754	1.575	2.210	0.964
	PointWOLF+FAT (Ours)	0.902	**0.993**	0.558	0.487	1.372	0.589	1.411	1.759	0.775
	WOLFMix	0.880	1.149	0.986	0.560	2.096	0.605	1.155	1.854	0.789
	WOLFMix+FAT (Ours)	0.882	**0.945**	0.726	0.491	1.691	0.520	1.048	1.498	0.644
PCT	Vanilla Training	0.930	0.925	1.042	0.870	0.872	0.528	1.000	0.780	1.385
	RSMix	0.925	0.660	1.116	0.614	1.106	0.488	0.522	0.302	0.473
	RSMix+FAT (Ours)	0.925	**0.604**	1.093	0.354	1.106	0.427	0.531	0.308	0.411
	PointWOLF	0.923	0.846	0.428	0.788	0.979	0.504	1.130	1.040	1.051
	PointWOLF+FAT (Ours)	0.923	**0.785**	0.465	0.415	1.096	0.556	1.217	0.953	0.796
	WOLFMix	0.922	0.585	0.442	0.788	0.989	0.444	0.546	0.319	0.564
	WOLFMix+FAT (Ours)	0.920	**0.570**	0.572	0.326	1.351	0.444	0.560	0.325	0.415
GDANet	Vanilla Training	0.934	0.892	0.981	0.839	0.830	0.794	0.894	0.871	1.036
	RSMix	0.927	0.680	1.205	0.873	1.000	0.484	0.531	0.281	0.385
	RSMix+FAT (Ours)	0.929	**0.617**	1.153	0.427	1.021	0.504	0.531	0.285	0.396
	PointWOLF	0.919	0.870	0.405	1.111	0.915	1.032	1.121	0.688	0.815
	PointWOLF+FAT (Ours)	0.925	**0.807**	0.428	0.522	0.915	0.831	1.159	1.058	0.735
	WOLFMix	0.920	0.601	0.428	0.937	0.968	0.540	0.589	0.298	0.444
	WOLFMix+FAT (Ours)	0.930	**0.537**	0.530	0.418	1.138	0.460	0.527	0.281	0.404

Table 3: Quantitative results of FAT and its variants. FAT w/o low-frequency has a lower mCE for high-frequency corruptions such as "Jitter", while FAT w/o high-frequency has a lower mCE for low-frequency corruptions such as "scale". FAT w/o Advprop has a higher mCE but much worse OA. Compared with these methods, FAT achieves the lowest mCE.

	Method	OA ↑	**mCE** ↓	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
PointNet	Vanilla Training	0.907	1.422	1.902	0.642	1.266	0.500	1.072	2.980	1.593
	FAT w/o low-frequency	0.890	1.306	1.614	0.373	1.734	0.504	1.193	2.627	1.098
	FAT w/o high-frequency	0.906	1.317	1.702	0.519	1.234	0.452	1.043	2.851	1.415
	FAT w/o frequency-division	0.904	1.310	1.679	0.516	1.351	0.444	1.106	2.817	1.255
	FAT w/o Advprop	0.885	1.263	1.470	0.411	1.926	0.500	1.164	2.461	0.909
	FAT	0.902	**1.237**	1.553	0.370	1.606	0.448	1.097	2.583	1.004

4.4 Ablation study

In this section, we conduct ablation study among our proposed Frequency Adversarial Training (FAT), as well as FAT variants: FAT w/o low-frequency, FAT w/o high-frequency, FAT w/o frequency-division, and FAT w/o Advprop. FAT w/o low-frequency generates only high-frequency adversarial samples, while FAT w/o high-frequency generates only low-frequency adversarial samples. FAT w/o frequency-division randomly generates adversarial samples within a certain frequency range, without dividing the high and low frequency bands. FAT w/o Advprop does not use the AdvProp training framework [72]. We compare these methods in Table 3 based on mean corruption error (mCE), clean overall accuracy (OA), and corruption error (CE) measurements for each corruption type.

Compared with other methods, FAT w/o low-frequency has a lower mCE for high-frequency corruptions such as "Jitter", while FAT w/o high-frequency has a lower mCE for low-frequency corruptions such as "scale". As discussed in Sec. 3.3, this is because adversarial training on high/low frequencies reduces the high/low frequency sensitivity, thus improving robustness to high/low-frequency corruptions. The performance of FAT w/o frequency-division falls between FAT w/o low-frequency and FAT w/o high-frequency. Although FAT w/o Advprop has a better mCE, its clean overall accuracy (OA) is worse than the other methods due to mutual interference between samples from different distributions, which may cause potential performance degradation. Compared with these methods, FAT achieves the lowest mCE, showing the effectiveness of our algorithm. More experimental results can be found in Appendix B.

https://doi.org/10.52202/079017-2177

5 Conclusion

In this paper, we study the robustness of 3D point cloud recognition models under common corruptions. We focus on the frequency domain to analyze the underlying structure of point clouds and common corruptions. Through graph Fourier transform (GFT), we identify a correlation between the corruption robustness and the model sensitivity to different frequency bands. Motivated by the analysis, we propose Frequency Adversarial Training (FAT), an adversarial training method based on frequency-domain adversarial examples to improve the corruption robustness of 3D point cloud recognition models. Extensive experiments demonstrate that the proposed method significantly improves the corruption robustness of various point cloud models, and can be integrated with other data augmentation techniques to achieve the state-of-the-art performance.

Limitation and broader impact. A limitation of our proposed method is that it reduces the clean accuracy a bit, e.g., FAT reduces the clean accuracy of DGCNN by 0.1%, PointNet by 0.5%, PCT by 1.0%, and GDANet by 0.6%. This may be caused by the inherent trade-off between accuracy and robustness [88]. Additionally, despite the complexity in implementation, FAT does not affect the efficiency of model inference, ensuring unhindered deployment of well-trained models in practical applications. The robustness of 3D point cloud recognition under corruptions is a severe problem towards safe and reliable 3D perception. Our work proposes an effective method to solve this issue, which does not have any negative social impact.

Acknowledgments

This work is supported by NKRDP grant No.2018YFA0704705, NSFC grants No.62276149 and No.12288201, and grant GJ0090202. Y. Dong is supported by the China National Postdoctoral Program for Innovative Talents. The authors thank anonymous referees for their valuable comments. L. Yu is supported by CAS Project for Young Scientists in Basic Research, Grant No.YSBR-040, ISCAS New Cultivation Project ISCAS-PYFX-202201, and ISCAS Basic Research ISCAS-JCZD-202302.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

[2] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, 2019.

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[4] Alvin Chan, Yew-Soon Ong, and Clement Tan. How does frequency bias affect the robustness of neural image classifiers against common corruption and adversarial perturbations? *arXiv preprint arXiv:2205.04533*, 2022.

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[7] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 330–345. Springer, 2020.

[8] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional

neural networks with octave convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3435–3444, 2019.

[9] Shuyu Cheng, Yibo Miao, Yinpeng Dong, Xiao Yang, Xiao-Shan Gao, and Jun Zhu. Efficient black-box adversarial attacks via bayesian optimization guided by a function prior. In *Forty-first International Conference on Machine Learning*, 2024.

[10] Wenda Chu, Linyi Li, and Bo Li. Tpc: Transformation-specific smoothing for point cloud models. *arXiv preprint arXiv:2201.12733*, 2022.

[11] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017.

[12] Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11513–11521. IEEE, 2020.

[13] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2019.

[14] Xiang Gao, Wei Hu, and Guo-Jun Qi. Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2020.

[15] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[16] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, pages 3809–3820. PMLR, 2021.

[17] 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*, 7:187–199, 2021.

[18] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 241–257. Springer, 2020.

[19] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[20] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[21] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15335–15344, 2022.

[22] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J Kim. Point cloud augmentation with weighted local transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 548–557, 2021.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee. Regularization strategy for point cloud via rigidly mixed sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15900–15909, 2021.

 https://doi.org/10.52202/079017-2177

[25] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6378–6387, 2020.

[26] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018.

[27] Binbin Liu, Jinlai Zhang, and Jihong Zhu. Boosting 3d adversarial attacks with attacking on frequency. *IEEE Access*, 10:50974–50984, 2022.

[28] Daizong Liu and Wei Hu. Imperceptible transfer attack and defense on 3d point cloud classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[29] Daizong Liu, Wei Hu, and Xin Li. Point cloud attacks in graph spectral domain: When 3d geometry meets graph signal processing. *arXiv preprint arXiv:2207.13326*, 2022.

[30] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5239–5248, 2019.

[31] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019.

[32] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

[33] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[34] Yibo Miao, Yinpeng Dong, Jun Zhu, and Xiao-Shan Gao. Isometric 3d adversarial examples in the physical world. In *Advances in Neural Information Processing Systems*, volume 35, pages 19716–19731, 2022.

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

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

[37] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022.

[38] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5376–5385, 2020.

[39] Mohammed H Rasheed, Omar M Salih, Mohammed M Siddeq, and Marcos A Rodrigues. Image compression based on 2d discrete fourier transform and matrix minimization algorithm. *Array*, 6:100024, 2020.

[40] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[41] Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. In *International Conference on Machine Learning*, pages 18559–18575. PMLR, 2022.

[42] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 53–69. Springer, 2020.

[43] Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs: Graph fourier transform. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6167–6170. IEEE, 2013.

[44] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018.

[45] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[46] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[47] Jiachen Sun, Karl Koenig, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of 3d point cloud classification under adaptive attacks. *arXiv preprint arXiv:2011.11922*, 2020.

[48] 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. *Advances in Neural Information Processing Systems*, 34:15498–15512, 2021.

[49] Jiachen Sun, Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, Dan Hendrycks, Jihun Hamm, and Z Morley Mao. A spectral view of randomized smoothing under common corruptions: Benchmarking and improving certified robustness. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 654–671. Springer, 2022.

[50] Jiachen Sun, Weili Nie, Zhiding Yu, Z Morley Mao, and Chaowei Xiao. Pointdp: Diffusion-driven purification against adversarial attacks on 3d point cloud recognition. *arXiv preprint arXiv:2208.09801*, 2022.

[51] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*, 2022.

[52] Jiachen Sun Sun, Yulong Cao Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *USENIX Security Symposium (Usenix Security'20)*, 2020.

[53] 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.

[54] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):1–28, 2015.

[55] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. Rgcnn: Regularized graph cnn for point cloud segmentation. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 746–754, 2018.

[56] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.

 https://doi.org/10.52202/079017-2177

[57] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 954–962, 2020.

[58] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13716–13725, 2020.

[59] Puru Vaish, Shunxin Wang, and Nicola Strisciuglio. Fourier-basis functions to bridge augmentation gap: Rethinking frequency augmentation in image classification. *arXiv preprint arXiv:2403.01944*, 2024.

[60] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017.

[61] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8684–8694, 2020.

[62] Jie Wang, Lihe Ding, Tingfa Xu, Shaocong Dong, Xinli Xu, Long Bai, and Jianan Li. Sample-adaptive augmentation for point cloud recognition against real-world corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14330–14339, 2023.

[63] Ruibin Wang, Yibo Yang, and Dacheng Tao. Art-point: Improving rotation robustness of point cloud classifiers via adversarial rotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14371–14380, 2022.

[64] 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*, pages 2569–2578, 2018.

[65] 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):1–12, 2019.

[66] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[67] 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*, pages 1912–1920, 2015.

[68] Ziyi Wu, Yueqi Duan, He Wang, Qingnan Fan, and Leonidas J Guibas. If-defense: 3d adversarial point cloud defense via implicit function based restoration. *arXiv preprint arXiv:2010.05272*, 2020.

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

[70] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 915–924, 2021.

[71] Chenxi Xiao and Juan Wachs. Triangle-net: Towards robustness in point cloud learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 826–835, 2021.

[72] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

[73] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021.

[74] Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3056–3064, 2021.

[75] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020.

[76] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5589–5598, 2020.

[77] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *Advances in neural information processing systems*, 32, 2019.

[78] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2019.

[79] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018.

[80] Mingyang Yi, Lu Hou, Jiacheng Sun, Lifeng Shang, Xin Jiang, Qun Liu, and Zhiming Ma. Improved ood generalization via adversarial training and pretraing. In *International Conference on Machine Learning*, pages 11987–11997, 2021.

[81] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32, 2019.

[82] Lijia Yu, Shuang Liu, Yibo Miao, Xiao-Shan Gao, and Lijun Zhang. Generalization bound and new algorithm for clean-label backdoor attack. In *Forty-first International Conference on Machine Learning*, 2024.

[83] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 186–194, 2018.

[84] Xiangyu Yue, Bichen Wu, Sanjit A Seshia, Kurt Keutzer, and Alberto L Sangiovanni-Vincentelli. A lidar point cloud generator: from a virtual world to autonomous driving. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 458–464, 2018.

[85] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[86] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

 https://doi.org/10.52202/079017-2177

[87] Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons. In *International Conference on Machine Learning*, pages 12368–12379, 2021.

[88] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482, 2019.

[89] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[90] Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujin Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification. *Neurocomputing*, 505:58–67, 2022.

[91] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1598–1606, 2019.

[92] Boxuan Zhong, He Huang, and Edgar Lobaton. Reliable vision-based grasping target recognition for upper limb prostheses. *IEEE Transactions on Cybernetics*, 2020.

[93] 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/CVF International Conference on Computer Vision*, pages 1961–1970, 2019.

[94] Shenchen Zhu, Yue Zhao, Kai Chen, Bo Wang, Hualong Ma, et al. {AE-Morpher}: Improve physical robustness of adversarial objects against {LiDAR-based} detectors via object reconstruction. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 7339–7356, 2024.

[95] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Wei Li, Yuexin Ma, Hongsheng Li, Ruigang Yang, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[96] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. Can we use arbitrary objects to attack lidar perception in autonomous driving? In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1945–1960, 2021.

[97] Yifan Zhu, Yibo Miao, Yinpeng Dong, and Xiao-Shan Gao. Toward availability attacks in 3d point clouds. In *International Conference on Machine Learning*, 2024.

A Proof

Theorem 1. *If f satisfies that: $f(x) \in [0, M]$ for all x, $|f(\mathcal{F}^{-1}(\mathcal{F}(x) + \alpha)) - f(x)| \leq \epsilon$ for all x and $\|\alpha\|_p \leq \delta$, then for any distribution P_o and P_a satisfying that $Was^p(P_o, P_a) := (\inf_{u \in \Pi(P_o, P_a)} \mathbb{E}_{(x,z)\sim u}[\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p])^{1/p} \leq \eta$, where $\eta < \delta$, then, with probability $1 - \gamma$, we have:*

$$\mathbb{E}_{z\sim P_o}[f(z)] - \frac{1}{m}\sum_{i=1}^{m} f(x_i) \leq \epsilon\left(1 - \frac{\eta^p}{\delta^p} - \sqrt{\frac{\ln(4/\gamma)}{2m}}\right) + \frac{\eta^p}{\delta^p}M + 4M\sqrt{\frac{\ln(4/\gamma)}{2m}}, \tag{A.1}$$

where $\{x_i\}_{i=1}^m$ are i.i.d. samples from P_a.

Proof. Assume u is a joint distribution of P_o and P_a, such that $(\mathbb{E}_{(x,z)\sim u}[\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p])^{1/p} \leq \eta$. Firstly, by Markov inequality, we have that:

$$\begin{aligned} & P_{(x,z)\sim u}(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p \geq \delta) \\ = \; & P_{(x,z)\sim u}(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p \geq \delta^p) \\ \leq \; & \frac{\mathbb{E}_{(x,z)\sim u}[\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p]}{\delta^p} \\ \leq \; & \frac{\eta^p}{\delta^p}. \end{aligned} \tag{A.2}$$

Then, we have that:

$$\mathbb{E}_{(x,z)\sim u}[I(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p \leq \delta)] = P_{(x,z)\sim u}(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p \leq \delta) \geq 1 - \frac{\eta^p}{\delta^p}. \tag{A.3}$$

Now, let $\{(x_i^u, z_i^u)\}_{i=1}^m$ be i.i.d. sampled from distribution u. Then, by Hoeffding inequality, we have that:

(1): with probability $1 - \gamma/4$, there are

$$\begin{aligned} & \tfrac{1}{m}\textstyle\sum_{i=1}^m I(\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p \leq \delta) \\ \geq \; & \mathbb{E}_{(x,z)\sim u}[I(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p \leq \delta)] - \sqrt{\tfrac{ln(4/\gamma)}{2m}} \\ \geq \; & 1 - \tfrac{\eta^p}{\delta^p} - \sqrt{\tfrac{ln(4/\gamma)}{2m}}, \end{aligned} \tag{A.4}$$

which indicates that there are at least $m(1 - \frac{\eta^p}{\delta^p} - \sqrt{\frac{ln(4/\gamma)}{2m}})$ number of $i \in [m]$ makes that $\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p \leq \delta$;

(2): with probability $1 - \gamma/4$, there are

$$-\frac{1}{m}\sum_{i=1}^{m} f(z_i^u) + \mathbb{E}_{z\sim P_o}[f(z)] = -\frac{1}{m}\sum_{i=1}^{m} f(z_i^u) + \mathbb{E}_{(x,z)\sim u}[f(z)] \leq M\sqrt{\frac{ln(4/\gamma)}{2m}}; \tag{A.5}$$

(3): with probability $1 - \gamma/4$, there are

$$\frac{1}{m}\sum_{i=1}^{m} f(x_i^u) - \mathbb{E}_{x\sim P_a}[f(x)] = \frac{1}{m}\sum_{i=1}^{m} f(x_i^u) - \mathbb{E}_{(x,z)\sim u}[f(x)] \leq M\sqrt{\frac{ln(4/\gamma)}{2m}}; \tag{A.6}$$

Let $\{x_i\}_{i=1}^m$ are i.i.d. samples from distribution P_a, then, by Hoeffding inequality, we have that:

(4): with probability $1 - \gamma/4$, there are

$$-\frac{1}{m}\sum_{i=1}^{m} f(x_i) + \mathbb{E}_{x\sim P_a}[f(x)] \leq M\sqrt{\frac{ln(4/\gamma)}{2m}}; \tag{A.7}$$

https://doi.org/10.52202/079017-2177

So, with probability $1-\gamma$ makes that (1), (2), (3) and (4) stand, at this times, we can estimate the $\mathbb{E}_{z\sim P_o}[f(z)] - \frac{1}{m}\sum_{i=1}^{m} f(x_i)$, there are:

$$\begin{aligned}
& \mathbb{E}_{z\sim P_o}[f(z)] \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^{m} f(z_i^u) + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^{m} f(x_i^u) + |f(x_i^u) - f(z_i^u)| + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^{m} f(x_i^u) + \epsilon I(\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p \leq \delta) + MI(\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p > \delta) + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^{m} f(x_i^u) + \epsilon(1 - \frac{\eta^p}{\delta^p} - \sqrt{\frac{ln(4/\gamma)}{2m}}) + (\frac{\eta^p}{\delta^p} + \sqrt{\frac{ln(4/\gamma)}{2m}})M + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \mathbb{E}_{x\sim P_a}[f(x)] + \epsilon(1 - \frac{\eta^p}{\delta^p} - \sqrt{\frac{ln(4/\gamma)}{2m}}) + (\frac{\eta^p}{\delta^p} + \sqrt{\frac{ln(4/\gamma)}{2m}})M + 2M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^{m} f(x_i) + \epsilon(1 - \frac{\eta^p}{\delta^p} - \sqrt{\frac{ln(4/\gamma)}{2m}}) + (\frac{\eta^p}{\delta^p} + \sqrt{\frac{ln(4/\gamma)}{2m}})M + 3M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
=\ & \tfrac{1}{m}\textstyle\sum_{i=1}^{m} f(x_i) + \epsilon(1 - \frac{\eta^p}{\delta^p} - \sqrt{\frac{ln(4/\gamma)}{2m}}) + M\frac{\eta^p}{\delta^p} + 4M\sqrt{\frac{ln(4/\gamma)}{2m}}.
\end{aligned} \tag{A.8}$$

We get our conclusion. □

Our generalization bound can also be extended to Lipschitz neural networks, which are a class of networks with global Lipschitz constants [11, 87].

Corollary A.1. *If f satisfies that: $f(x) \in [0, M]$ for all $x \in [0,1]^n$, $|f(\mathcal{F}^{-1}(\mathcal{F}(x)+\alpha)) - f(x)| \leq \epsilon\|\alpha\|_p$ for all $x \in [0,1]^n$ and α, then for any distribution P_o and P_a in $[0,1]^n$ satisfying that $Was^p(P_o, P_a) := (\inf_{u\in\Pi(P_o,P_a)} \mathbb{E}_{(x,z)\sim u}[\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p])^{1/p} \leq \eta$, then, with probability $1-\gamma$, we have:*

$$\mathbb{E}_{z\sim P_o}[f(z)] - \frac{1}{m}\sum_{i=1}^{m} f(x_i) \leq \epsilon(\eta + v\eta\sqrt{\frac{ln(4/\gamma)}{2m}}) + \frac{M}{v^p} + 3M\sqrt{\frac{ln(4/\gamma)}{2m}}, \tag{A.9}$$

where $\{x_i\}_{i=1}^m$ are i.i.d. samples from P_a, v is any real number greater than 1.

Proof. Assuming u is a joint distribution of P_o and P_a, and makes that $(\mathbb{E}_{(\mathcal{F}(x),\mathcal{F}(z))\sim u}[\|x - z\|_p^p])^{1/p} \leq \eta$. Firstly, by Markov inequality, we have that:

$$\begin{aligned}
& P_{(x,z)\sim u}(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p \geq v\eta) \\
=\ & P_{(x,z)\sim u}(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p \geq (v\eta)^p) \\
\leq\ & \frac{\mathbb{E}_{(x,z)\sim u}[\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p]}{(v\eta)^p} \\
\leq\ & \frac{\eta^p}{(v\eta)^p} = (1/v)^p.
\end{aligned} \tag{A.10}$$

Then, we have that:

$$\mathbb{E}_{(x,z)\sim u}(I(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p \leq v\eta)) = P_{(x,z)\sim u}(\|\mathcal{F}(x) - \mathcal{F}(z)\|_p \leq v\eta) \geq 1 - \frac{1}{v^p}. \tag{A.11}$$

Now, let $\{(x_i, z_i)\}_{i=1}^m$ are i.i.d. samples from distribution u. Then, by Hoeffding inequality, we have that:

(1): with probability $1 - \gamma/4$, there are

$$\begin{aligned}
& \tfrac{1}{m}\textstyle\sum_{i=1}^{m} I(\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p \leq v\eta)\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p \\
\leq\ & \mathbb{E}_{(x,z)\sim u}[I(\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p \\
\leq\ & v\eta)\|\mathcal{F}(x) - \mathcal{F}(z)\|_p] + v\eta\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & (\mathbb{E}_{(x,z)\sim u}[\|\mathcal{F}(x) - \mathcal{F}(z)\|_p^p])^{1/p} + v\eta\sqrt{\frac{ln(4/\gamma)}{2m}} \\
=\ & \eta + v\eta\sqrt{\frac{ln(4/\gamma)}{2m}};
\end{aligned} \tag{A.12}$$

(2): with probability $1 - \gamma/4$, there are

$$-\frac{1}{m}\sum_{i=1}^{m} f(z_i^u) + \mathbb{E}_{z\sim P_o}[f(z)]| = -\frac{1}{m}\sum_{i=1}^{m} f(z_i^u) + \mathbb{E}_{(x,z)\sim u}[f(z)] \leq M\sqrt{\frac{ln(4/\gamma)}{2m}}; \tag{A.13}$$

(3): with probability $1-\gamma/4$, there are

$$\frac{1}{m}\sum_{i=1}^{m} f(x_i^u) - \mathbb{E}_{x\sim P_a}[f(x)] = \frac{1}{m}\sum_{i=1}^{m} f(x_i^u) - \mathbb{E}_{(x,z)\sim u}[f(x)] \leq M\sqrt{\frac{ln(4/\gamma)}{2m}}; \tag{A.14}$$

Let $\{x_i\}_{i=1}^m$ are i.i.d. samples from distribution P_a, then, by Hoeffding inequality, we have that:

(4): with probability $1-\gamma/4$, there are

$$-\frac{1}{m}\sum_{i=1}^{m} f(x_i) + \mathbb{E}_{x\sim P_a}[f(x)] \leq M\sqrt{\frac{ln(4/\gamma)}{2m}}; \tag{A.15}$$

So, with probability $1-\gamma$ makes that (1), (2), (3) and (4) stand, at this times, we can estimate the $\mathbb{E}_{z\sim P_o}[f(z)] - \frac{1}{m}\sum_{i=1}^m f(x_i)$, there are:

$$\begin{aligned}
& \mathbb{E}_{z\sim P_o}[f(z)] \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^m f(z_i^u) + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^m f(x_i^u) + |f(x_i^u) - f(z_i^u)| + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^m f(x_i^u) + \epsilon\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p I(\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p \leq v\eta) \\
& + MI(\|\mathcal{F}(x_i^u) - \mathcal{F}(z_i^u)\|_p > v\eta) + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^m f(x_i^u) + \epsilon(\eta + v\eta\sqrt{\frac{ln(4/\gamma)}{2m}}) + \frac{M}{v^p} + M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \mathbb{E}_{x\sim P_a}[f(x)] + \epsilon(\eta + v\eta\sqrt{\frac{ln(4/\gamma)}{2m}}) + \frac{M}{v^p} + 2M\sqrt{\frac{ln(4/\gamma)}{2m}} \\
\leq\ & \tfrac{1}{m}\textstyle\sum_{i=1}^m f(x_i) + \epsilon(\eta + v\eta\sqrt{\frac{ln(4/\gamma)}{2m}}) + \frac{M}{v^p} + 3M\sqrt{\frac{ln(4/\gamma)}{2m}}.
\end{aligned} \tag{A.16}$$

We get our conclusion. □

B Supplementary experimental results

In this section, we provide more experimental results. All of the experiments are conducted on NVIDIA Tesla V100 GPUs.

B.1 The performance in terms of mOA and RmCE

In this section, we present full results for corruption overall accuracy (mOA) and relative mCE (RmCE) [41]. The mOA is computed as the average OA over all corruptions. The RmCE quantifies the performance drop compared to a clean test set. We adopt the official baseline DGCNN and

Table B.1: Quantitative results of vanilla training, adversarial training and our proposed Frequency Adversarial Training (FAT) on the ModelNet-C test set. Our proposed FAT outperforms all other methods in terms of corruption overall accuracy (mOA), which demonstrates the effectiveness of FAT for improving corruption robustness.

	Method	OA ↑	**mOA** ↑	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
DGCNN	Vanilla Training	0.926	0.764	0.785	0.684	0.906	0.752	0.793	0.705	0.725
	Adv Training	0.925	0.790	0.781	0.816	0.902	0.753	0.772	0.743	0.761
	FAT (Ours)	0.925	**0.815**	0.807	0.857	0.907	0.769	0.799	0.772	0.791
PointNet	Vanilla Training	0.907	0.658	0.591	0.797	0.881	0.876	0.778	0.121	0.562
	Adv Training	0.904	0.673	0.602	0.822	0.879	0.889	0.777	0.148	0.591
	FAT (Ours)	0.902	**0.717**	0.666	0.883	0.849	0.889	0.773	0.238	0.724
PCT	Vanilla Training	0.930	0.781	0.776	0.725	0.918	0.869	0.793	0.770	0.619
	Adv Training	0.919	0.778	0.776	0.877	0.899	0.774	0.753	0.673	0.693
	FAT (Ours)	0.920	**0.798**	0.783	0.891	0.898	0.791	0.744	0.731	0.747
GDANet	Vanilla Training	0.934	0.789	0.789	0.735	0.922	0.803	0.815	0.743	0.715
	Adv Training	0.926	0.781	0.761	0.840	0.903	0.770	0.764	0.686	0.742
	FAT (Ours)	0.928	**0.810**	0.749	0.871	0.913	0.803	0.770	0.807	0.756

https://doi.org/10.52202/079017-2177

Table B.2: Quantitative results of vanilla training, adversarial training and our proposed Frequency Adversarial Training (FAT) on the ModelNet-C test set. Our proposed FAT outperforms all other methods in terms of relative mCE (RmCE), which demonstrates the effectiveness of FAT for improving corruption robustness.

	Method	OA ↑	**RmCE ↓**	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
DGCNN	Vanilla Training	0.926	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Adv Training	0.925	0.914	1.021	0.450	1.150	0.989	1.150	0.824	0.816
	FAT (Ours)	0.925	**0.746**	0.837	0.281	0.899	0.897	0.947	0.692	0.667
PointNet	Vanilla Training	0.907	1.488	2.241	0.455	1.300	0.178	0.970	3.557	1.716
	Adv Training	0.904	1.393	2.142	0.339	1.250	0.086	0.955	3.421	1.557
	FAT (Ours)	0.902	**1.334**	1.674	0.079	2.650	0.075	0.970	3.005	0.886
PCT	Vanilla Training	0.930	0.884	1.092	0.847	0.600	0.351	1.030	0.724	1.547
	Adv Training	0.919	0.929	1.014	0.174	1.000	0.833	1.248	1.113	1.124
	FAT (Ours)	0.920	**0.853**	0.972	0.120	1.098	0.741	1.323	0.855	0.861
GDANet	Vanilla Training	0.934	0.865	0.753	0.822	0.600	0.895	0.864	1.090	1.028
	Adv Training	0.926	0.970	1.170	0.355	1.150	0.897	1.218	1.086	0.915
	FAT (Ours)	0.928	**0.795**	1.270	0.236	0.750	0.718	1.188	0.548	0.856

Table B.3: Quantitative results of vanilla training and our proposed Frequency Adversarial Training (FAT) on the ModelNet40-C test set. Our proposed FAT outperforms other methods in terms of mCE, which demonstrates the effectiveness of FAT for improving corruption robustness.

	Method	OA ↑	**mCE ↓**	Uni.	Gauss.	Impluse	Upsamp.	Back.	Occlu.	LiDAR	Den.Inc.	Den.Dec.	Cutout	Rotate	Shear	FFD	RBF	Inv.RBF
DGCNN	VT	0.926	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	FAT	0.925	**0.782**	0.687	0.601	0.551	0.580	0.764	0.952	0.912	0.881	0.919	0.863	0.808	0.824	0.817	0.787	0.785
PointNet	VT	0.907	1.157	0.850	0.871	1.168	0.733	1.763	0.883	0.677	0.739	0.671	0.778	1.930	2.105	1.626	1.277	1.274
	FAT	0.902	**1.009**	0.763	0.731	0.833	0.631	1.689	0.904	0.694	0.710	0.633	0.730	1.688	1.807	1.328	1.012	0.979
PCT	VT	0.929	0.959	0.828	0.839	1.568	0.908	1.091	0.956	0.947	0.837	0.829	0.942	0.949	0.956	0.943	0.892	0.898
	FAT	0.920	**0.699**	0.584	0.535	0.519	0.553	0.604	0.953	0.739	0.653	0.757	0.753	0.872	0.858	0.814	0.647	0.644
GDANet	VT	0.934	0.869	0.950	0.995	0.912	0.864	0.801	0.987	0.829	0.698	0.690	0.777	0.947	0.914	0.883	0.900	0.885
	FAT	0.928	**0.764**	0.645	0.604	0.502	0.579	0.508	0.998	0.817	0.776	0.805	0.819	1.029	0.979	0.888	0.758	0.753

initially calculate the relative corruption error (RCE) for a given corruption type i by averaging over 5 severity levels: $\mathrm{RCE}_i = \frac{\sum_{l=1}^{5}(\mathrm{OA}_{\mathrm{clean}} - \mathrm{OA}_{i,l})}{\sum_{l=1}^{5}(\mathrm{OA}_{\mathrm{clean}}^{\mathrm{DGCNN}} - \mathrm{OA}_{i,l}^{\mathrm{DGCNN}})}$, where $\mathrm{OA}_{\mathrm{clean}}$ is the overall accuracy on the clean test set. Subsequently, we compute the relative mean corruption error (RmCE) by averaging over the 7 corruption types: $\mathrm{RmCE} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{RCE}_i$. In Tables B.1 and B.2, we compare different methods based on the mOA and RmCE metrics, confirming that our proposed FAT enhances the model's out-of-distribution generalization ability.

B.2 The performance on the ModelNet40-C

In this section, we evaluate the corruption robustness on ModelNet40-C [51]. The ModelNet40-C dataset is specifically designed to assess the network robustness against prevalent point cloud corruptions. It consists of 15 different corruption types, including "Uniform", "Gaussian", "Impulse", "Upsampling", "Background", "Occlusion", "LiDAR", "Local Density Inc", "Local Density Dec", "Cutout", "Rotation", "Shear", "FFD", "RBF", and "Inv RBF". Each type of corruption has 5 severity

Table B.4: Quantitative results of the performance of FAT when integrated with data augmentation strategy in terms of mCE and $\mathrm{ER}_{\mathrm{cor}}$ on the ModelNet40-C test set. In previous studies [51], PCT with CutMix-R achieves the best robustness with the 0.635 mCE and 0.163 $\mathrm{ER}_{\mathrm{cor}}$. However, training GDANet with the combination of our proposed FAT with WOLFMix achieves the new state-of-the-art performance, with the impressive 0.555 mCE and **0.147 $\mathrm{ER}_{\mathrm{cor}}$**.

Method	OA ↑	**mCE ↓**	Noise	Density	Trans.	**$\mathrm{ER}_{\mathrm{cor}}$ ↓**	Noise	Density	Trans.
PCT+CutMix-R	0.928	0.635	0.469	0.669	0.766	0.163	0.105	0.271	0.112
PCT+WOLFMix	0.922	0.627	0.580	0.673	0.629	0.158	0.118	0.267	0.090
PCT+WOLFMix+FAT (Ours)	0.920	**0.583**	0.454	0.636	0.658	**0.151**	0.097	0.261	0.094
GDANet+WOLFMix	0.920	0.669	0.709	0.685	0.612	0.171	0.137	0.289	0.087
GDANet+WOLFMix+FAT (Ours)	0.930	**0.555**	0.450	0.627	0.588	**0.147**	0.094	0.263	0.084

Table B.5: Quantitative results of vanilla training, adversarial training, DUP Defense and our proposed Frequency Adversarial Training (FAT) on the ScanObjectNN-C test set. Our proposed FAT outperforms all other methods in terms of mean corruption error (mCE), which demonstrates the effectiveness of FAT for improving corruption robustness.

	Method	OA ↑	**mCE** ↓	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
DGCNN	Vanilla Training	0.858	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Adv Training	0.843	1.062	1.146	0.778	1.097	0.873	0.960	1.185	1.396
	DUP Defense	0.832	1.029	1.195	0.833	1.157	1.197	1.214	0.773	0.834
	FAT (Ours)	0.856	**0.933**	0.968	0.815	0.959	0.852	0.950	0.959	1.026
PointNet	Vanilla Training	0.739	1.354	1.610	0.884	1.427	0.786	1.264	1.487	2.022
	Adv Training	0.725	1.334	1.532	0.844	1.403	0.873	1.333	1.474	1.881
	DUP Defense	0.712	1.348	1.717	0.825	1.555	1.157	1.480	0.829	1.875
	FAT (Ours)	0.734	**1.254**	1.393	0.796	1.465	0.844	1.264	1.259	1.759
PointNext	Vanilla Training	0.873	0.921	0.995	1.079	0.803	0.807	0.942	0.944	0.875
	Adv Training	0.870	0.901	0.991	1.027	0.803	0.833	0.929	0.912	0.809
	DUP Defense	0.859	0.901	0.980	1.046	0.826	0.748	0.973	0.923	0.809
	FAT (Ours)	0.875	**0.877**	0.998	0.916	0.791	0.786	0.867	0.938	0.840

Table B.6: Quantitative results of vanilla training, adversarial training, DUP Defense and our proposed Frequency Adversarial Training (FAT) on the PointNeXt on ModelNet-C. Our proposed FAT outperforms all other methods in terms of mean corruption error (mCE), which demonstrates the effectiveness of FAT for improving corruption robustness.

	Method	OA ↑	**mCE** ↓	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
PointNeXt	Vanilla Training	0.932	0.856	1.460	1.297	0.904	0.847	0.957	0.251	0.276
	Adv Training	0.924	0.834	1.593	0.716	1.025	0.876	1.144	0.230	0.251
	DUP Defense	0.919	0.840	1.461	0.838	1.192	0.715	1.188	0.224	0.262
	FAT (Ours)	0.930	**0.781**	1.412	0.692	0.986	0.827	1.082	0.230	0.241

levels. In Table B.3, we compare different methods on the ModelNet40-C test set, confirming that our proposed FAT enhances the OOD generalization ability of the model.

In Table B.4, we show the performance of FAT when integrated with data augmentation strategy in terms of mCE and ER_{cor}. In previous studies [51], PCT with CutMix-R achieves the best robustness with the 0.635 mCE and 0.163 ER_{cor}. However, training GDANet with the combination of our proposed FAT with WOLFMix achieves a new state-of-the-art performance, with the impressive 0.555 mCE and 0.147 ER_{cor}.

B.3 The performance on the ScanObjectNN-C

We further conduct experiments on the ScanObjectNN dataset, which is collected by LiDAR sensors and represents more realistic conditions under real-world scenarios [2, 58, 96, 94]. The experimental settings and evaluation metrics on ScanObjectNN-C [62] are consistent with those on ModelNet-C. The results are shown in Table B.5. It can be seen that our FAT generally leads to lower mCE on ScanObjectNN-C. The experimental results on ScanObjectNN-C further validate the generalizability and applicability of our FAT under real-world conditions.

B.4 The performance on the PointNeXt

We further conduct experiments for the updated point cloud model PointNeXt [37]. The results in Table B.5 and Table B.6 demonstrate that FAT achieves consistent performance on the advanced network architecture PointNeXt, similar to observations on PointNet and more. Our FAT outperforms all other methods in terms of mCE. This indicates that FAT's performance is largely independent of the underlying model architecture, making it applicable to both traditional and modern networks.

B.5 The performance in combination with AdaptPoint

We further conduct experiments for comparison with AdaptPoint [62] on ScanObjectNN-C. AdaptPoint follows the official experimental settings. The results are shown in Table B.7. It is evident that incorporating FAT achieves a lower mCE, indicating its superiority.

https://doi.org/10.52202/079017-2177

Table B.7: Quantitative results of combining FAT with AdaptPoint on ScanObjectNN-C. Compared with single AdaptPoint, the combination of FAT and AdaptPoint achieves a better mCE.

	Method	OA ↑	**mCE** ↓	Rotate	Jitter	Scale	Drop-G	Drop-L	Add-G	Add-L
PointNet	AdaptPoint	0.743	1.256	1.359	0.875	1.519	0.676	1.112	1.448	1.804
	AdaptPoint+FAT (Ours)	0.744	**1.196**	1.370	0.823	1.446	0.690	1.125	1.220	1.701
PointNext	AdaptPoint	0.885	0.783	0.767	1.030	0.810	0.508	0.628	0.911	0.824
	AdaptPoint+FAT (Ours)	0.885	**0.761**	0.748	0.948	0.833	0.521	0.648	0.829	0.802

https://doi.org/10.52202/079017-2177

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our paper supports the claims made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed limitations in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-2177

Justification: We have provided the full set of assumptions in every theorem and made a complete proof in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided reproductive details in Section 4.1 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided our codes in the supplemental matrial.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided experimental details in Section 4.1 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: For fair comparison, we do not provide error bars because there are many baseline methods, it is computationally expensive to reproduce all of these methods.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-2177

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Justification: We have provided them in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed them in Appendix 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use open-source dataset and models in our paper, and have cited the original paper of these dataset and models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-2177

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.