Sample-adaptive Augmentation for Point Cloud Recognition Against Real-world Corruptions

Jie Wang[1], **Lihe Ding**[1], **Tingfa Xu**[1,†], **Shaocong Dong**[1], **Xinli Xu**[1], **Long Bai**[2], **Jianan Li**[1,†]
[1] Beijing Institute of Technology [2] The Chinese University of Hong Kong
https://github.com/Roywangj/AdaptPoint

Abstract

Robust 3D perception under corruption has become an essential task for the realm of 3D vision. While current data augmentation techniques usually perform random transformations on all point cloud objects in an offline way and ignore the structure of the samples, resulting in over-or-under enhancement. In this work, we propose an alternative to make sample-adaptive transformations based on the structure of the sample to cope with potential corruption via an auto-augmentation framework, named as AdaptPoint. Specially, we leverage a imitator, consisting of a Deformation Controller and a Mask Controller, respectively in charge of predicting deformation parameters and producing a per-point mask, based on the intrinsic structural information of the input point cloud, and then conduct corruption simulations on top. Then a discriminator is utilized to prevent the generation of excessive corruption that deviates from the original data distribution. In addition, a perception-guidance feedback mechanism is incorporated to guide the generation of samples with appropriate difficulty level. Furthermore, to address the paucity of real-world corrupted point cloud, we also introduce a new dataset ScanObjectNN-C, that exhibits greater similarity to actual data in real-world environments, especially when contrasted with preceding CAD datasets. Experiments show that our method achieves state-of-the-art results on multiple corruption benchmarks, including ModelNet-C, our ScanObjectNN-C, and ShapeNet-C.

1. Introduction

3D vision has gained attention for its potential applications in robotics and autonomous driving. Existing methods [14, 15, 22, 16] primarily focused on clean data [26, 1] with few corruptions, whereas practical applications often entail numerous corruptions. Consequently, these models exhibit poor performance in such scenarios. As such, improving the robustness of point cloud models against corruption remains a crucial but challenging problem.

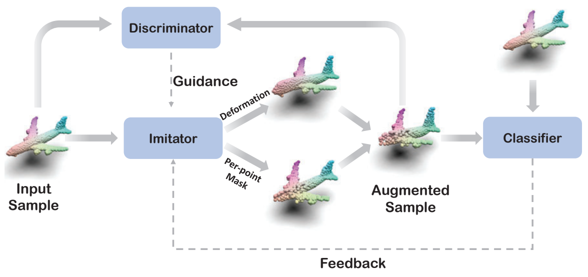

Figure 1. **Workflow of AdaptPoint**. The AdaptPoint framework involves a co-evolving strategy where the imitator, discriminator, and point cloud classifier are jointly trained. During the training process, the imitator is trained to transform clean data into realistic corruptions while receiving feedback and guidance from the classifier and discriminator, respectively.

To enhance the model's resistance to corruption, a frequently employed technique in prior studies is data augmentation [7, 9, 18]. Despite their effectiveness, these methods are subject to two common limitations. Firstly, these offline methods fail to consider the structure of the classifier, which may lead to the generation of samples that are not beneficial for enhancing model robustness. Additionally, they often overlook intrinsic features of the data and apply random transformations to all samples, resulting in augmentations that are either overly difficult or overly simplistic.

To tackle the issues at hand, we introduce a new auto-augmentation framework, AdaptPoint, that is specifically tailored to generate augmented samples that consider the inherent structure of input point cloud. At the heart of our proposed method lies the sample-adaptive imitator network, which is designed to generate point cloud that are representative of real-world scenarios. It first extracts local geometric features from the input point cloud, resulting in a subset of points that encode these features. Subsequently, the imitator network is divided into two branches, namely the **Deformation Controller** and the **Mask Controller**. The former module, is tasked with modifying the point cloud struc-

†Correspondence to: Jianan Li and Tingfa Xu.

ture by incorporating the interactions between local regions and predicting the deformation parameters for specific local areas. On the other hand, the Mask Controller, is responsible for selectively removing certain points by leveraging the overall structural information of the point cloud and generating a per-point mask. Finally, the deformation and mask parameters are fed into a corruption simulator to generate a corrupted point cloud. The design of the deformation and mask mechanisms endows the imitator with the ability to simulate most real-world corruptions.

The sample-adaptive imitator designed has an impressive capability to generate samples that closely resemble real-world corruptions. The following problem to be considered is how to provide sufficient guidance to the imitator network to ensure: i) the generated samples conform to the real point cloud distribution; and ii) generating the samples with appropriate difficulty for classifier training. To achieve these objectives, we introduce a discriminator and a classifier. The discriminator is used to distinguish between the distributions of clean and augmented samples, thereby preventing the generation of excessive corruption that deviates from the original data distribution. On the other hand, the classifier is employed to provide classification error feedback to guide the imitator network in generating samples with an appropriate level of difficulty. These three key components are integrated into the proposed AdaptPoint framework, as illustrated in Fig. 1. By jointly optimizing the imitator with a point cloud discriminator and classifier, our framework progressively generates corrupted data, thereby enhancing the model's robustness against corruption.

Current datasets for point cloud corruption evaluation [18, 17] are largely limited to CAD models, which fail to capture the variety of real-world corruptions encountered in practice and may hinder research progress in point cloud robustness. To bridge this gap, we present a new point cloud corruption dataset, namely ScanObjectNN-C, which is curated by collecting real-world point clouds from ScanObjectNN [20] dataset, along with corresponding annotations and five types of corruptions: "Jitter","Drop Global/Local", "Add Global/Local", "Scale", and "Rotate" [18].

The effectiveness of AdaptPoint is validated through extensive experimentation on diverse datasets such as ModelNet-C [18], ShapeNet-C[17], and our more challenging ScanObjectNN-C. Our proposed AdaptPoint achieves state-of-the-art results on these datasets, indicating its capability to alleviate the negative influence of data corruptions on point cloud perception performance. Furthermore, our method achieves competitive results in defending point cloud attacks, indicating its generality and effectiveness in handling different types of corruption scenarios.

The main contributions of our work are as follows:

- We present a new auto-augmentation framework, AdaptPoint, which is tailored for point cloud recognition in the presence of real-world corruptions.
- We propose a novel sample-adaptive imitator network that can simulate various types of realistic corruptions by leveraging the intrinsic characteristics of input.
- We construct a real-world point cloud corruption dataset ScanObjectNN-C, which can facilitate research on real-world point cloud corruption.
- We establish new state-of-the-art results on several corruption benchmarks.

2. Related Work

Deep Learning on Point Cloud Corruption. Despite the efficacy of existing models in point cloud perception [14, 15, 12, 16, 24, 29, 22], they remain vulnerable to corruption. Recent research has highlighted the increasing importance of mitigating point cloud corruption [18, 17, 7, 9]. Kim et al.[7] proposed PointWolf, which enhances point cloud model robustness using non-rigid deformation, while Lee et al.[9] proposed RSMix, which generates mixed samples by rigidly transforming two point clouds. Ren et al.[18] proposed WOLFMix, which combines PointWOLF and RSmix to further enhance model robustness. Additionally, Ren et al.[17] proposed an attention-based backbone RPC for point cloud corruption. Although these methods have shown improvement in model robustness against corruption to some extent, they may not perform well on real-world datasets. Our proposed online auto-augmentation framework, AdaptPoint, provides a solution to this vulnerability.

Data Augmentation on Point Cloud. Data augmentation has become widely accepted in the deep learning community for enhancing the generalization performance of deep neural networks. However, traditional data augmentation techniques such as random scaling, rotation, and jitter have limitations in improving the capability of models for point cloud analysis. To overcome this, advanced data augmentation techniques have been proposed in recent studies, such as PointAugment [10], PointMixup [2], and PointWOLF [7]. Nevertheless, PointMixup [2] and PointWOLF [7] primarily perform pre-defined transformations, and PointAugment [10] only learns sample-wise global transformations while neglecting the significance of local geometry in point cloud. In this paper, we introduce a novel auto-augmentation framework for combating corruptions that can adjust augmentations according to both global and local features of point clouds, as well as the current capability of the model.

2D and 3D Robustness Benchmarks. Deep neural networks are known to be susceptible to corruption in data. To improve the robustness of these networks, ImageNet-C [6] was introduced, which corrupts the test set of ImageNet [3] using simulated corruptions such as motion blur. Similarly, two benchmarks for robust point cloud recognition under corruptions were introduced with ModelNet-C [18] and

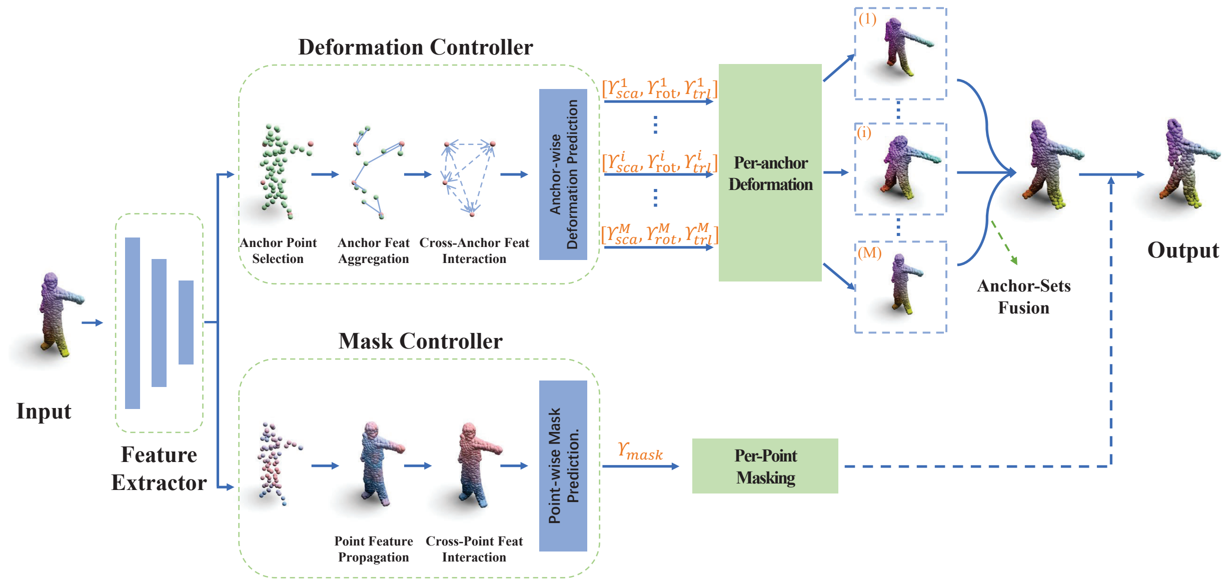
Deformation Controller
Anchor Point Selection
Anchor Feat Aggregation
Cross-Anchor Feat Interaction
Anchor-wise Deformation Prediction
$[\gamma^1_{sca}, \gamma^1_{rot}, \gamma^1_{trl}]$
$[\gamma^i_{sca}, \gamma^i_{rot}, \gamma^i_{trl}]$
$[\gamma^M_{sca}, \gamma^M_{rot}, \gamma^M_{trl}]$
Per-anchor Deformation
(1)
(i)
(M)
Anchor-Sets Fusion
Output
Input
Feature Extractor
Mask Controller
Point Feature Propagation
Cross-Point Feat Interaction
Point-wise Mask Prediction.
γ_{mask}
Per-Point Masking

Figure 2. **The overall architecture of sample-adaptive Imitator.** It initially leverages a Feature Extractor to extract local geometric features from the input point cloud. Subsequently, the Deformation Controller and Mask Controller modules are employed to predict deformation parameters specific to multiple anchors and generate a mask for the entire point cloud, respectively. A Corruption Simulator is then employed to apply corresponding transformations and masks to the input point cloud, resulting in an augmented point cloud.

ShapeNet-C [17]. However, these two datasets are largely limited to CAD models, which fail to capture the variety of real-world corruptions encountered in practice and may hinder research progress in point cloud robustness. To bridge this gap, we present a novel point cloud corruption dataset, namely ScanObjectNN-C.

3. Method

Given an input point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$, our AdaptPoint aims to produce a corresponding corrupted point cloud $\mathcal{P}' \in \mathbb{R}^{N\times 3}$ based on the structural characteristics of $\mathcal{P}$ for improving the robustness of the classifier against corruptions. This is made possible through introducing a corruption imitator, which is jointly optimized with a discriminator alongside the classifier. Specifically, the imitator simulates common operations of corruptions in most real-life scenarios to ensure the plausibility. The discriminator helps weaken the generation of excessive corruption which is far away from raw data distribution. The classifier provides instant feedback and guide imitator to generate samples with suitable difficulty. These three key components are integrated into a self-contained framework and co-evolve as the training proceeds. With raw clean data as input, AdaptPoint progressively generates and refines corresponding corrupted data and robust classifier, which in turn reinforces all components in the loop.

3.1. Sample-adaptive Imitator

The proposed sample-adaptive imitator workflow is presented in Fig. 2, consisting of four key modules: i) the Feature Extractor, implemented by PointNet++, captures local geometric features from the input point cloud $\mathcal{P}$, and outputs N' sampled points $\mathcal{G} \in \mathbb{R}^{N'\times 3}$ with corresponding features $F = \{f_j | j = 1, ..., N'\} \in \mathbb{R}^{N'\times C}$; ii) the Deformation Controller that responsible for capturing the interactions between selected anchor points and predicting deformation parameters specific to each anchor point, (iii) the Mask Controller that takes into consideration the overall structural information of the point cloud and predicts a per-point mask, and (iv) the Corruption Simulator that utilizes the predicted parameters and mask to apply transformations and occlusions to the input point cloud.

3.1.1 Deformation Controller

The Deformation Controller takes point cloud $\mathcal{P}$ and extracted feature F as input, captures region features and enables multiple region transformations, thus generating deformation for point cloud by a convex combination of multiple locally-defined transformations. To this end, we first select M anchor points from $\mathcal{P}$ and aggregate local features centered on these anchor points, then establish the inner connections between them, finally predict deformation parameters, specifically for scaling $\gamma_{sca} = \{\gamma^i_{sca} | i = 1, ..., M\} \in \mathbb{R}^{M\times 3}$, rotation $\gamma_{rot} = \{\gamma^i_{rot} | i = 1, ..., M\} \in \mathbb{R}^{M\times 3}$ and translation $\gamma_{trl} = \{\gamma^i_{trl} | i = 1, ..., M\} \in \mathbb{R}^{M\times 3}$.

Per-anchor Feature aggregation. In this step, a set of M anchor points $\mathcal{D} \in \mathbb{R}^{M\times 3}$ are initially selected from $\mathcal{P}$, using the Farthest Point Sampling (FPS) algorithm, serving as the basis for the subsequent anchor-wise deformation process, where each anchor is independently transformed. For the i-th anchor, we select its K-nearest neighbors from $\mathcal{G}$ and learns its local feature representation:

$$h_i = \underset{j=1,\cdots,K}{\mathcal{A}} \{\Phi(f_{i,j})\} \in \mathbb{R}^{1\times C}, \tag{1}$$

where $f_{i,j}$ is the j-th neighbor point feature of i-th anchor point. $\Phi(\cdot)$ is implemented by multi-layer perceptron (MLP). $\mathcal{A}(\cdot)$ denotes a symmetric function, *e.g.*, max pooling, to aggregate encoded point features. As a result, anchor features $H = \{h_i | i = 1, ..., M\} \in \mathbb{R}^{M\times C}$ are obtained.
Cross-anchor Feature Interaction. In pursuit of maintaining the cohesion of the comprehensive deformation, the mutual correlation of local transformations corresponding to each anchor set holds substantial relevance. A juxtaposition of myriad disconnected transformations may instigate a considerable disruption to the input. To counteract such unwelcome implications, it is essential to highlight the integration of anchor sets, along with building their connection.

Therefore, a transformer network with multi-head-attention is utilized to maintain structural information of anchor sets. Given anchor features H with anchor coordinates $\mathcal{D}$, we first get query (Q), key (K) and value (V) as:

$$Q, K, V = HW^Q,\ HW^K,\ HW^V, \tag{2}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{C\times C}$ are learnable linear projections. Next, we obtain updated anchor features $H' = \{h'_i | i = 1, ..., M\} \in \mathbb{R}^{M\times C}$ as:

$$H' = MHA(Q, K, V, PE(\mathcal{D})), \tag{3}$$

where MHA represents a multi-head attention [21] and $PE(\cdot)$ denotes the position embedding.
Anchor-wise Deformation Prediction. Predicting deformation requires not only connection between anchor sets, but also global information to complement and constrain the overall shape of the object. Hence, the global structural information of all the anchors is encoded as

$$g = \underset{i=1,\cdots,M}{\mathcal{A}} \{\phi(h_i)\} \in \mathbb{R}^{1\times C}, \tag{4}$$

where ϕ is a point-wise feature embedding and $\mathcal{A}$ denotes max pooling, respectively.

The subsequent step involves predicting deformation parameters for each anchor point.This is achieved by employing multiple linear layers with different activation functions for each deformation type:

$$\gamma_k^j = \alpha_k(\psi_k([h'_j, g])),\ k \in \{sca, rot, trl\} \tag{5}$$

where $[\cdot,\cdot]$ denotes the concatenation operation. α is the activation function and the mapping function ψ is a point-wise feature transformations, such as linear projections or MLPs. Specifically, Sigmoid activation function is used for scaling and Tanh activation function for rotation and translation.

3.1.2 Mask Controller

Occurrence of corruption leads to alterations not only in the geometric structure of the point cloud, but also results in partial omissions of certain parts to some extent. The Mask Controller accounts for the overall structure of the input point cloud and the interactions between all points, further predicting a point-wise mask to emulate the absence of points in the real world. The presented module operates by initially getting point-wise feature at raw resolution, $E \in \mathbb{R}^{N\times C}$, from $F \in \mathbb{R}^{N'\times C}$ through simple trilinear interpolation. It subsequently performs feature interaction across distinct points. The final output of this module is a point-wise mask $\gamma_{mask} = \left\{\gamma_{mask}^j | j = 1, ..., N\right\} \in \mathbb{R}^{N\times 1}$.
Cross-point Feature Interaction. For improved point-wise mask prediction and mitigation of large-scale omissions that may distort the output samples, an approach that accounts for point-to-point connectivity is essential. We employ multi-head attention, similar as that used in the deformation controller, to model the relationships between points and get the updated point features $E' = \left\{E'_j | j = 1, ..., N\right\} \in \mathbb{R}^{N\times C}$.
Point-wise Mask Prediction. The accurate prediction of a point-wise Mask necessitates the utilization of both inter-point characteristics and holistic object structure. To this end, we first extract a global point feature $z \in \mathbb{R}^{1\times C}$ by following the similar process as used in the deformation controller. Then we utilize the complementary local and global point features to predict the parameters of the mask:

$$\gamma_{mask}^j = \sigma(\rho([E'_j, z])) \in \mathbb{R}^{1\times 1}, \tag{6}$$

where ρ denotes a point-wise feature embedding learned by a MLP. σ represents a normalization function implemented by gumbel-softmax.

3.1.3 Corruption Simulator

This module utilizes the predicted deformation parameters $\{\gamma_{sca}, \gamma_{rot}, \gamma_{trl}\}$ and mask γ_{mask} to perform transformations input on point cloud. The resultant point cloud exhibits structural modifications and partial point loss, effectively simulating the characteristics of real-world corrupted point clouds.
Per-anchor deformation. To introduce local deformations in point clouds, we utilize a combination of scaling, rotation, and translation transformations with parameters predicted by the Deformation Controller module. Firstly, we normalize the input point cloud $\mathcal{P}$ using each anchor points in $\mathcal{D}$ to obtain a set of normalized point clouds $\mathcal{Q} = \{q_i \in \mathbb{R}^{N\times 3} | i = 1, ..., M\} \in \mathbb{R}^{M\times N\times 3}$. Then, for the i-th normalized point cloud q_i, we apply a transformation based on the predicted deformation parameters, resulting in a deformed point cloud set $\mathcal{Q}' = \left\{q'_i \in \mathbb{R}^{N\times 3} | i = 1, ..., M\right\} \in$

$\mathbb{R}^{M\times N\times 3}$. Specifically, we apply the scaling, rotation, and translation transformations as follows:

$$q'_i = (q_i \cdot \gamma^i_{sca})R(\gamma^i_{rot}) + \gamma^i_{trl}, \tag{7}$$

where $R(\cdot)$ represents the rotation matrix. This process results in diverse local transformations for the deformed point cloud sets, with each set corresponding to a anchor.

Anchor-Sets Fusion. We assemble the set of locally deformed point clouds $\mathcal{Q}'$ into a whole point cloud $\overline{\mathcal{P}} \in \mathbb{R}^{N\times 3}$ through anchor-sets fusion strategy [7]. To smoothly interpolate the local transformations in the 3D space, we employ the Nadaraya-Watson kernel regression inspired by the operation of *Smooth deformations* [7].

Per-point Masking. This step involves applying the predicted mask $\gamma_{mask} \in \mathbb{R}$ to the fused point cloud $\overline{\mathcal{P}}$. The masked point cloud $\mathcal{P}'$ is obtained as

$$\mathcal{P}' = \overline{\mathcal{P}} \odot \gamma_{mask} \in \mathbb{R}^{N\times 3}. \tag{8}$$

Here, $\odot$ denotes element-wise multiplication. The purpose of mask is to imitate the effect of sensor noise or data missing that can occur during data acquisition. The resulting $\mathcal{P}'$ is taken as the final output augmented point cloud.

3.2. Learning Objectives

We present the AdaptPoint framework that leverages a sample-adaptive imitator along with a discriminator and classifier. The loss function comprises of the discriminator loss, $\mathcal{L}_{adv}$, and the feedback loss, $\mathcal{L}_{feed}$, combined to form the complete loss function:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda\mathcal{L}_{feed}, \tag{9}$$

where λ is a balancing weight, set as 1 in our experiments.

Feedback Loss. To ensure the augmented data utilized for model training is more arduous than the original data yet not impractical due to excessive difficulty, inspired by PoseAug [4], we introduce a feedback loss function. The feedback loss is designed to constrain the difference between the classification loss on augmented and original data within a proper range as follows:

$$\mathcal{L}_{feed} = |1 - exp[\mathcal{L}_{\mathcal{C}}(\mathcal{P}') - \beta\mathcal{L}_{\mathcal{C}}(\mathcal{P})]|. \tag{10}$$

where $\mathcal{L}_{\mathcal{C}}$, the cross-entropy loss between predicted labels and ground truths, is used as the classifier loss metric. Here, the parameter $\beta > 1$ controls the difficulty level of augmentation, ensuring that the value of $\mathcal{L}_{\mathcal{C}}(\mathcal{P}')$ stays within a certain range relative to $\mathcal{L}_{\mathcal{C}}(\mathcal{P})$. As the classifier's ability improves over time, we gradually increase the value of β to elevate the degree of augmentation.

Discriminator loss. Merely striving for augmentations that maximize errors may spawn unfeasible training instances that infringe upon the structure of the point cloud, consequently inflicting detriment upon model performance. To alleviate such a concern, we incorporate a point cloud discriminator module within the point cloud's structure to facilitate the training of the augmentor, thereby confirming the plausibility of the augmented samples without undermining the diversity. Specially, we adopt PointNet++[15] as the discriminator. The adversarial loss is:

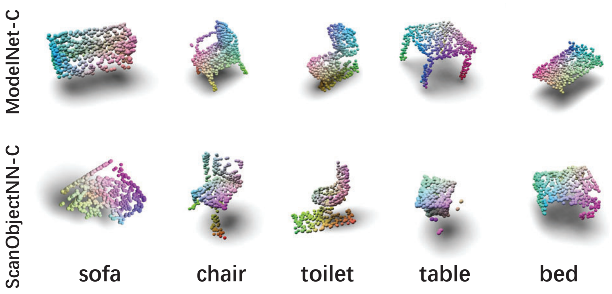

Figure 3. Some samples in ModelNet-C and ScanObjectNN-C.

$$\mathcal{L}_{adv} = \mathbb{E}_{x\sim\mathbb{P}(\mathcal{P})}[log(D(I(x)))], \tag{11}$$

where x is sampled from the probability distribution of the input point cloud data $\mathcal{P}$. $I(\cdot)$ denotes the mapping function learned by the imitator. $D(I(x))$ is the discriminator's estimate of the probability that the augmented data sample $I(x)$ is clean.

4. ScanObjectNN-C Dataset

Current point cloud corruption ModelNet-C [18] and ShapeNet-C [17] are both from current point cloud perception dataset [26, 1], where the objects in that are simulated rather than scanned from real-world objects, may not adequately reflect a classifier's ability in realistic scenarios.

To evaluate real-world robustness, We build a point cloud corruption dataset, dubbed as ScanObjectNN-C, which is more in line with the real-world corruption. We collect point cloud from the test set of the most challenging variant of ScanObjectNN [20], PB-T50-RS, then process them by seven types of corruptions, "Jitter", "Drop Global/Local", "Add Global/Local", "Scale" and "Rotate", with five levels of severity to provide a comprehensive robustness evaluation. Specifically, objects from the ModelNet-C are derived from CAD models, whereas samples from ScanObjectNN-C are captured from real-world environments, often exhibiting occlusions, partiality, and complex backgrounds. The presented Fig. 3 evidences the distinction between objects from the ModelNet-C dataset and samples from ScanObjectNN-C, indicating that the corrupting influence present in the ScanObjectNN-C more closely approximates the realistic conditions encountered in the real world, in comparison to ModelNet-C.

5. Experiments

We evaluate AdaptPoint on several point cloud corruption datasets and also test it for point cloud attack defense.

Table 1. Classification results of mCE(%) on ModelNet-C.

Method	mCE(↓)	Sca	Jit	Drop-G	Drop-L	Add-G	Add-L	Rot
DGCNN [23]	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
PointNet [14]	142.2	126.6	**64.2**	50.0	107.2	298.0	159.3	190.2
RSCNN [11]	113.0	107.4	117.1	80.6	151.7	71.2	115.3	147.9
SimpleView [5]	104.7	87.2	71.5	124.2	135.7	98.3	84.4	131.6
GDANet [30]	89.2	83.0	83.9	79.4	89.4	87.1	103.6	98.1
CurveNet [28]	92.7	87.2	72.5	71.0	102.4	134.6	100.0	80.9
PAConv [29]	110.4	90.4	146.5	100.0	100.5	108.5	129.8	96.7
PointNet++ [15]	107.2	87.2	117.7	64.1	180.2	61.4	99.3	140.5
+ PointWolf [7]	82.5	**81.9**	135.1	67.3	130.4	43.1	68.4	51.2
+ Rsmix [9]	86.3	89.4	164.9	48.4	73.9	26.1	32.7	168.4
+ Wolfmix [18]	64.1	94.7	137.0	46.0	61.4	29.8	29.5	50.7
+ AdaptPoint	63.7	109.6	102.2	36.7	66.7	30.5	40.0	60.5
PointNeXt [16]	85.6	90.4	129.7	84.7	95.7	25.1	27.6	146.0
+ PointWolf [7]	79.5	88.3	144.9	113.7	103.9	26.4	28.0	51.2
+ Rsmix [9]	87.9	100.0	159.2	86.7	54.1	**23.1**	26.5	165.6
+ Wolfmix [18]	74.0	84.0	156.0	119.4	56.5	23.7	**25.1**	53.0
+ AdaptPoint	67.7	101.1	116.8	60.5	69.1	29.5	32.4	64.2
RPC [18]	86.3	84.0	89.2	49.2	79.7	92.9	101.1	107.9
+PointWolf [7]	70.2	88.3	107.3	80.2	104.8	27.1	29.1	54.4
+Rsmix [9]	71.0	98.9	100.9	72.2	62.3	26.4	29.1	107.4
+Wolfmix [18]	60.1	101.1	96.8	42.3	**51.2**	33.2	48.0	**47.9**
+AdaptPoint	**56.5**	93.6	77.5	**36.3**	72.9	27.5	29.1	58.6

5.1. Results on ModelNet-C

Data and Setup. We train models on the clean ModelNet40 [26] dataset and evaluate them on the ModelNet-C [18] corruption test suite. The imitator and discriminator are optimized using the Adam optimizer [8] and learning rates of 0.0001 and 0.0004, respectively. Mean Corruption Error metric (mCE, %, ↓) is taken as the main evaluation matrix. More details are in the supplementary materials.

Results. In our experiments, PointNet++ [15] and RPC [18] are adopted as the baseline models. Tab. 1 presents the results of various methods on the robustness against corruption. Our proposed AdaptPoint outperforms all other methods and achieves the state-of-the-art mCE. The incorporation of AdaptPoint brings substantial improvements in model performance, as both PointNet++ [15] and RPC [18] equipped with our method achieve an mCE of 63.7% and 56.5%, respectively, surpassing models without AdaptPoint by a large margin of over 20%. Notably, our method improves the performance of nearly all categories, and exhibits the most significant improvements in drop-local and add-local, which evaluates the effectiveness of local deformation. Compared with PointWOLF [7] and Rsmix [9], our method achieves a significant increase in mCE for jitter, over 30% on both baselines, which bears testament to AdaptPoint's efficacy in amplifying the local geometric nuances of the point cloud, thereby effectuating minor yet consequential modifications to the local structure.

Visualization of Augmented Samples. Fig. 4 showcases the visualization of both clean and augmented data generated by two distinct techniques, PointWOLF and AdaptPoint. This figure reveals that the augmented samples generated by AdaptPoint exhibit superior diversity compared

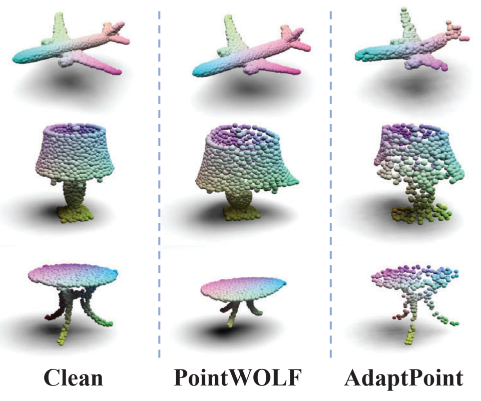

Figure 4. Examples of augmented samples by AdaptPoint.

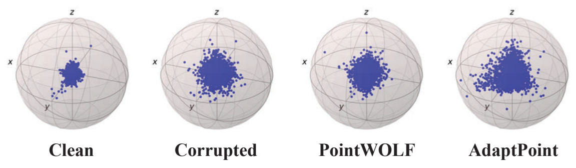

Figure 5. Visualization of data distribution of raw data from ModelNet40, corrupted data by a commonly employed corruption technique (i.e., Drop-Global), and augmented data from PointWOLF and AdaptPoint. Our AdaptPoint outperforms the other methods in terms of generating diverse results.

to those generated by PointWOLF, thereby aligning better with real-world scenarios of corruption. This signifies that our model induces a more intricate transformation in the geometry of the point cloud beyond just elementary distortions or absences. This advanced level of geometric modulation imparts robustness to the classifier model, as substantiated by the perceived enhancements in the quality of the produced samples.

Visualization of Data Distribution. To showcase the efficacy of AdaptPoint in augmenting data diversity, we conduct an analysis on the distribution of point cloud positions. As illustrated in Fig. 5, we compare the point position distributions of ModelNet40 with data corrupted by a commonly used corruption,(i.e., Drop-Global), data augmented by[7] and our method. Our findings reveal that the distribution in the original ModelNet40 dataset are concentrated near the origin of coordinates and have a limited range, which may explain why the models trained on this dataset exhibit poor generalization to corrupted scenarios. Furthermore, we observe that the point position distribution of the augmented data generated by [7] has a limited degree of divergence, indicating that the improvements made through the use of handcrafted rules are constrained. In contrast, our AdaptPoint method leverages a learnable augmentor to generate more diverse and realistic point positions, resulting in a greater degree of data augmentation.

Visualization of Salient Geometry. In order to gain deeper insights into the impact of the imitator on clean data, we conducted a comparative analysis of the features attended

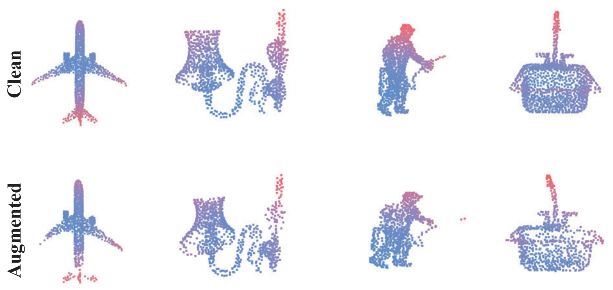

Figure 6. Visualization of salient geometry learned by the classifier. Points with high / low scores are in red / blue.

to by the classifier on clean and augmented data. Fig. 6 provides a visual representation of the focus of the classifier's first perception layer. We computed the average features of the points and utilized them to represent their contribution to the classifier. Points that contributed more to the classifier were assigned higher scores in color, with red indicating high scores and blue indicating low scores. The illustration exposes that, despite significant transformations in the morphology of the augmented data, it still maintains the crucial local prototypes and continues to supply the perception model with equivalent semantic information.

5.2. Results on ScanObjectNN-C

Data and Setup. Objects in ScanObjectNN are scanned from real-world objects, and performance on ScanObjectNN may better reflect classifier ability in realistic scenarios. We train models on ScanObjectNN dataset and evaluate them on ScanObjectNN-C corruption test suite. We train the model for 250 epochs using the Adam optimizer with a learning rate of 0.002.

Results. Tab. 2 presents the experimental results of our proposed method. Our method achieves mCE of 78.3% on PointNeXt-S, outperforming the current state-of-the-art method WOLFMix [18]. Across all models evaluated, our method achieved a lift of approximately 18% in average mCE performance, which significantly surpassed the previous state-of-the-art method's improvement of 5.6% in WOLFMix. Meanwhile, our method achieves overall accuracy (OA) of 88.45% on ScanObjectNN, which is a highly competitive result surpassing many advanced methods [16, 12]. These results demonstrate the generalizability of our proposed method and its potential for improving the performance of various models.

5.3. Results on Shapenet-C

Data and Setup. We conduct part segmentation experiment on ShapeNet-C [17]. We employed PointMAE [13] as the baseline and trained the model for 300 epochs with the AdamW optimizer. The learning rate was set to 0.0002.

Results. Tab. 3 demonstrates that our method achieves a score of 76.9% mCE and outperforms the state-of-the-art PointMAE method by a margin of 15.4% . It is worth mentioning that PointMAE is a MAE model, and the significant improvement provided by our AdaptPoint suggests its effectiveness in enhancing the performance of self-supervised models, beyond being limited to fully supervised models. In addition to its efficacy in classification, our proposed AdaptPoint augmentation technique showcases its potential on ShapeNet-C dataset by achieving favorable performance. Remarkably, our method achieved these results while preserving the integrity of the semantic information, attesting to its robustness and practicality. demonstrating superiority of our approach in downstream tasks

Table 2. Classification results of mCE(%) on ScanObjectNN-C.

Method	OA(↑)	mCE(↓)	Sca	Jit	D-G	D-L	A-G	A-L	Rot
DGCNN [23]	85.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
+PointWOLF	85.6	99.6	89.5	104.6	104.1	98.3	98.0	100.9	101.6
+RSMix	86.5	96.9	103.1	97.4	91.2	85.3	104.8	98.7	97.6
+WOLFMix	87.2	92.3	92.1	102.6	92.6	85.3	96.2	85.7	91.5
+AdaptPoint	84.4	90.2	90.6	107.5	72.3	73.0	93.3	93.3	101.4
PointNet++ [15]	86.2	96.9	89.7	110.3	55.0	127.7	94.7	90.5	110.7
+PointWOLF	86.6	96.4	84.0	108.7	70.5	156.6	**87.7**	90.9	76.1
+RSMix	87.3	91.9	89.0	100.7	55.6	99.0	94.6	89.1	114.9
+WOLFMix	87.5	87.8	**79.6**	109.0	64.2	117.7	88.1	**79.8**	76.2
+AdaptPoint	86.7	84.6	86.1	112.4	44.2	77.4	92.3	86.4	93.3
RPC [18]	74.7	132.6	131.7	107.3	145.5	130.5	114.2	158.7	140.2
+PointWOLF	71.0	141.5	132.9	113.1	142.1	133.7	135.1	173.2	160.5
+RSMix	65.7	146.9	149.3	**94.8**	127.0	133.2	153.6	208.8	161.3
+WOLFMix	79.7	120.2	108.5	117.4	122.3	107.2	123.3	124.6	138.1
+AdaptPoint	83.5	96.5	106.4	100.5	66.9	77.8	107.3	105.9	110.9
PointNeXt [16]	87.3	92.1	80.3	107.9	80.7	94.2	94.4	87.5	99.5
+PointWOLF	87.4	89.5	81.4	112.9	89.8	92.3	95.0	83.7	71.1
+RSMix	88.1	88.2	83.9	107.3	74.9	73.3	96.2	82.9	99.1
+WOLFMix	87.7	86.9	81.9	119.3	89.7	78.0	89.3	80.0	**70.0**
+AdaptPoint	**88.5**	**78.3**	81.0	103.0	**50.8**	**62.8**	91.1	82.4	76.7

Table 3. Segmenatation results of mCE (%) on ShapeNet-C

Method	mCE(↓)	Sca	Jit	Drop-G	Drop-L	Add-G	Add-L	Rot
DGCNN [23]	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
PointNet [14]	117.8	108.2	105.0	98.3	113.2	138.6	117.3	143.8
PointNet++ [15]	111.2	95.0	108.1	85.6	198.3	88.6	108.3	94.7
PAConv [29]	92.7	92.7	107.2	92.5	92.7	74.3	94.8	94.8
GDANet [30]	92.3	92.2	**101.2**	94.2	94.6	71.2	95.7	96.9
PT [33]	104.9	107.6	107.2	103.2	108.1	111.2	106.6	90.7
Point-MLP [12]	97.7	96.5	113.2	88.7	99.1	92.9	106.1	87.6
Point-BERT [32]	103.3	93.8	109.8	87.3	92.7	117.0	119.9	102.5
Point-MAE [13]	92.7	**90.8**	103.5	85.2	**88.2**	77.6	103.1	100.3
+AdaptPoint	**76.9**	95.2	104.8	**84.5**	92.1	**41.1**	**42.6**	**78.1**

5.4. Results on Point Cloud Attack Defense

Setup. Motivated by the remarkable performance of the AdaptPoint method against corruption, we explore its application in point cloud defense. We evaluate the performance of PointNet++ [15] trained using the AdaptPoint method on point perturbation attack [27], individual point adding attack [27], kNN attack [19] and point dropping attack [34].

Results. Based on the results presented in Tab. 5, it is observed that our AdaptPoint consistently outperforms all other defense methods for two-point dropping and adding attacks. The accuracy improvement can be as high as 4.5%

Table 4. Ablation studies

Feedback	Adv	mCE(↓)
✗	✓	82.1
✓	✗	83.2
✓	✓	**78.3**

(a) Effect of loss functions.

Deformation	Mask	mCE(↓)
✗	✗	92.1
✓	✗	86.4
✗	✓	87.1
✓	✓	**78.3**

(b) Effect of imitator components.

Anchor	mCE(↓)
2	79.9
4	**78.3**
8	80.2
16	81.4

(c) Effect of anchor number.

λ	mCE(↓)
0.5	78.7
1	**78.3**
2	79.0

(d) Impact of loss weight λ.

Table 5. Results on point cloud attack defense (OA, %).

Method	Perturb	Add-CD	Add-HD	kNN	Drop-100	Drop-200
NoDefense	-	7.24	6.59	-	80.19	68.96
SRS [31]	73.14	65.32	43.11	49.96	64.51	39.60
SOR [35]	77.67	72.90	72.41	61.35	74.16	69.17
SOR-AE [35]	78.73	73.38	71.19	78.73	76.66	68.23
Adv Training	20.03	12.27	10.06	8.63	80.39	67.14
DUP-Net [36]	80.63	75.81	72.45	74.88	76.38	72.00
IF-Defense [25]	**86.99**	80.19	76.09	**85.62**	84.56	79.09
AdaptPoint	86.75	**80.83**	**77.03**	77.67	**86.55**	**83.59**

in the Drop-200 attack, indicating the scalability and effectiveness of the approach. Even with the Perturb attack, which introduces very fine perturbations leading to a chaotic point cloud, AdaptPoint achieves the second-best performance, closely approaching the state-of-the-art defense method [25]. The above analysis leads to the conclusion that our method has a strong generalization ability across various point cloud attack algorithms.

5.5. Ablation Studies

PointNeXt [16] is taken as the classifier in ablations. More implementation details and ablations on hyperparameters are available in supplementary materials.

Effect of Feedback and Adversarial Loss. To ascertain the impact of feedback loss, we conducted an ablation study by removing feedback loss. Tab. 4a, unequivocally demonstrate that removing feedback loss led to a dramatic drop (4% ↓) in model performance, underscoring it critical importance. Besides, we also illustrate the effectiveness of plausibility guidance from the discriminator. The remove of Adversarial loss leads to 4.9% performance drop, demonstrating the crucial role of point cloud plausibility in enhancing classifier performance, with the discriminator playing a pivotal role in the process.

Effect of Deformation and Mask Controller. We conducted an investigation into the efficacy of the Deformation and Mask controller. Tab. 4b presents the results obtained by adding each component to a base pipeline. The performance gains achieved by incorporating each component alone were limited. Notably, combining both deformation and mask led to a substantial decrease in mCE, with the value reducing from 92.1% to 78.3%. This finding suggests that the incorporation of both deformation and mask significantly improves model robustness.

Effect of the Number of Anchors. We investigate the impact of anchor numbers of the deformation controller. As illustrated in Tab. 4c, we vary the number of anchors and analyze the resulting mCE scores. The results show that the performance of mCE is 81.4% when the number of weight matrices is 16, which is 3.1% higher than the performance achieved using 4 anchors. This observation can be attributed to the fact that too many anchors can negatively impact the overall deformation, therefore leading to a decrease in performance.

Impact of Feedback Loss Weight. Tab. 4d presents an empirical investigation of the impact of loss weight on the feedback loss in the context of AdaptPoint. The findings demonstrate that modifying the loss weight λ can lead to a harmful effect on model performance, due to the disruption of the delicate balance between the adversarial loss and the feedback loss. Specifically, increasing or decreasing the loss weight results in a drop in performance. Thus, maintaining a proper balance among all components in AdaptPoint is crucial, which further corroborates the efficacy of our imitator loss design.

Combination of corruption techniques. There exist two common techniques for point cloud data augmentation: predefined transformations like PointWOLF, and mix-up based sample generation like RSMix. WOLFMix combines both techniques, leveraging their strengths. Results in Tab. 1 in Sec. 5.1 and Tab. 2 in Sec. 5.2, involve all these techniques under both standalone and combined circumstances, offering a holistic comparison with our method.

6. Conclusion

In this paper, we present a novel auto-augmentation methodology, denominated AdaptPoint, specifically designed for point cloud recognition under real-world corruptions. The AdaptPoint approach leverages the benefits of a sample-adaptive imitator, enabling the simulation of real-world corruptions scenarios. Additionally, we introduce a real-world corruption dataset to facilitate the evaluation of point cloud recognition in the presence of real-world corruptions. Our experimental results, conducted on three benchmark datasets, manifest the superiority of our proposed approach.

Acknowledgments This work was financially supported by the National Natural Science Foundation of China (No. 62101032), the Postdoctoral Science Foundation of China (Nos. 2021M690015, 2022T150050), and Beijing Institute of Technology Research Fund Program for Young Scholars (No. 3040011182111).

References

[1] 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. 1, 5

[2] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In ECCV, pages 330–345. Springer, 2020. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248–255. Ieee, 2009. 2

[4] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In CVPR, pages 8575–8584, 2021. 5

[5] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In ICML, pages 3809–3820, 2021. 6

[6] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261, 2019. 2

[7] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J Kim. Point cloud augmentation with weighted local transformations. In ICCV, pages 548–557, 2021. 1, 2, 5, 6

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 6

[9] 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 CVPR, pages 15900–15909, 2021. 1, 2, 6

[10] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Pointaugment: an auto-augmentation framework for point cloud classification. In CVPR, pages 6378–6387, 2020. 2

[11] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In CVPR, 2019. 6

[12] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In ICLR, 2022. 2, 7

[13] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In ECCV, pages 604–621. Springer, 2022. 7

[14] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In CVPR, 2017. 1, 2, 6, 7

[15] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In NIPS, 2017. 1, 2, 5, 6, 7

[16] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In NIPS. 1, 2, 6, 7, 8

[17] Jiawei Ren, Lingdong Kong, Liang Pan, and Ziwei Liu. Pointcloud-c: Benchmarking and analyzing point cloud perception robustness under corruptions. preprint, 2022. 2, 3, 5, 7

[18] Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. arXiv preprint arXiv:2202.03377, 2022. 1, 2, 5, 6, 7

[19] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In AAAI, volume 34, pages 954–962, 2020. 7

[20] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In ICCV, 2019. 2, 5

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. NIPS, 30, 2017. 4

[22] Jie Wang, Jianan Li, Lihe Ding, Ying Wang, and Tingfa Xu. Papooling: Graph-based position adaptive aggregation of local geometry in point clouds. arXiv preprint arXiv:2111.14067, 2021. 1, 2

[23] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. TOG, 38(5):1–12, 2019. 6, 7

[24] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In CVPR, 2019. 2

[25] 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. 8

[26] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In CVPR, 2015. 1, 5, 6

[27] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In CVPR, pages 9136–9144, 2019. 7

[28] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In ICCV, 2021. 6

[29] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In CVPR, 2021. 2, 6, 7

[30] 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 AAAI, volume 35, pages 3056–3064, 2021. 6, 7

[31] Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. arXiv preprint arXiv:1902.10899, 2019. 8

[32] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In CVPR, pages 19313–19322, 2022. 7

[33] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In ICCV, 2021. 7

[34] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In ICCV, pages 1598–1606, 2019. 7

[35] Hang Zhou, Dongdong Chen, Jing Liao, Kejiang Chen, Xiaoyi Dong, Kunlin Liu, Weiming Zhang, Gang Hua, and Nenghai Yu. Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks. In CVPR, pages 10356–10365, 2020. 8

[36] 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 CVPR, pages 1961–1970, 2019. 8