Enhancing Sampling Protocol for Point Cloud Classification Against Corruptions

Chongshou Li[1,2] , **Pin Tang**[1,2] , **Tianrui Li**[1,2] , **Yuheng Liu**[3] , **Xinke Li***[4]
[1]School of Computing and Artificial Intelligence, Southwest Jiaotong University
[2]Engineering Research Center of Sustainable Urban Intelligent Transportation
Ministry of Education, Chengdu, China
[3]SWJTU-Leeds Joint School, Southwest Jiaotong University
[4]College of Computing, City University of Hong Kong
{lics, trli}@swjtu.edu.cn, tangpin1874@163.com, yuhengliu02@gmail.com, xinkeli@cityu.edu.hk

Abstract

Established sampling protocols for 3D point cloud learning, such as Farthest Point Sampling (FPS) and Fixed Sample Size (FSS), have long been relied upon. However, real-world data often suffer from corruptions, such as sensor noise, which violates the benign data assumption in current protocols. As a result, these protocols are highly vulnerable to noise, posing significant safety risks in critical applications like autonomous driving. To address these issues, we propose an enhanced point cloud sampling protocol, **PointSP**, designed to improve robustness against point cloud corruptions. PointSP incorporates key point reweighting to mitigate outlier sensitivity and ensure the selection of representative points. It also introduces a local-global balanced downsampling strategy, which allows for scalable and adaptive sampling while maintaining geometric consistency. Additionally, a lightweight tangent plane interpolation method is used to preserve local geometry while enhancing the density of the point cloud. Unlike learning-based approaches that require additional model training, PointSP is architecture-agnostic, requiring no extra learning or modification to the network. This enables seamless integration into existing pipelines. Extensive experiments on synthetic and real-world corrupted datasets show that PointSP significantly improves the robustness and accuracy of point cloud classification, outperforming state-of-the-art methods across multiple benchmarks.

1 Introduction

In the rapidly evolving field of 3D data perception via deep learning [Qi *et al.*, 2017a; Qi *et al.*, 2017b; Guo *et al.*, 2020], point cloud sampling serves as a critical component in the standard learning and recognition pipeline [Hu *et al.*, 2020; Qian *et al.*, 2022; Yu *et al.*, 2022; Zhang *et al.*, 2022b]. Following the legacy of pioneer works [Qi *et al.*, 2017a; Qi *et al.*, 2017b], existing sampling protocols are primarily designed and optimized for clean data, without taking into account corruptions. However, due to the high complexity of real-world, point cloud data are almost always incomplete and with noise in practice [Ren *et al.*, 2022], posing threats to 3D deep learning applications. For example, noisy background points or slight perturbations generated by inaccurate processing or sensor error can significantly decrease the deep model performance [Ren *et al.*, 2022; Sun *et al.*, 2022]. Such performance drops can lead to serious safety consequences, especially in critical 3D applications like autonomous driving. Therefore, it is necessary to rethink and redesign point cloud sampling protocols with a focus on robustness against corruptions to ensure reliable 3D deep learning in real-world conditions.

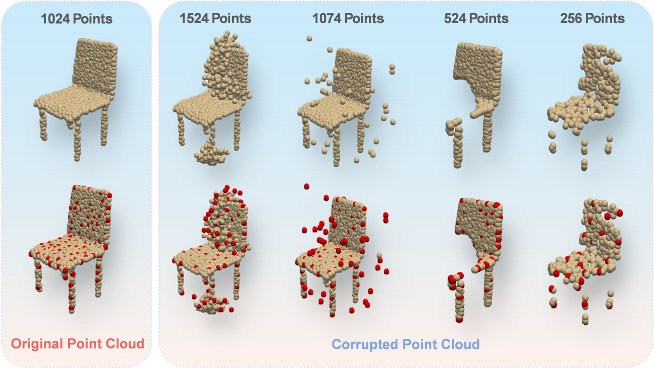

Figure 1: The first row presents the original point clouds, while the second row highlights the sampled key points, with those selected by farthest point sampling (FPS) shown in red. The existing standard sampling protocol is not optimized for corrupted point cloud in practice. As a result, a standard-trained PointNet classifies them as Chair, Vase, Table, Mantel, and Sofa. Towards this issue, we propose to enhance the protocol by revising FPS into **new key points selections** and integrating **full points resampling** into process.

One crucial limitation of established sampling protocols is that they are sub-optimal under the corrupted data distribution. For instance, the protocol samples a fixed number of points in data preparation, namely, Fixed Sample Size (FSS) [Qi *et al.*, 2017a]. This convention overlooks the facts that point clouds in the real world naturally vary in size and density. These varied sizes are even obvious in particular corruptions such as occlusions and density-related noise [Sun

*Corresphonding author

et al., 2022; Ren *et al.*, 2022]. The misclassfication results on various corrupted data are illustrated in Figure 1. Another aspect is that the widely used Farthest Point Sampling (FPS) [Eldar *et al.*, 1997] for key points selection is especially vulnerable to outliers due to its inherent basis of Euclidean distance and sensitivity to sparse points [Yan *et al.*, 2020]. Several works have considered updating a specific step to deal with this issue, like PointASNL [Yan *et al.*, 2020] and ADS [Hong *et al.*, 2023]. The learning-based methods put extra effort into module training and may be potentially overfitting. Overall, none of them propose a comprehensive and alternative solution to overcome the sampling protocol limitations.

To overcome these limitations, we propose an enhanced point cloud sampling protocol, **PointSP**, by revising key points selection and full points resampling.The implementation of the proposed protocol involves randomizing the sampling during training and processing noisy point clouds during inference. To achieve this, the key point sampling process assisted by point reweighting is applied to ensure that potential outliers are not captured. The point weight is named as isolation rate evaluating the extent of local isolation for a point. Moreover, the proposed full points resampling randomizes the sample size during training; and restores insufficient point clouds in the inference stage. We achieve an local-global-balanced downsampling offering a continuous spectrum between local and global sampling. Inspired by shape-invariant perturbation [Huang *et al.*, 2022], we realize an lightweight upsampling via a tangent plane interpolation technique that enhances the density of point cloud data while preserving local geometry.Overall, the enhanced sampling protocol is learning-free thus straightforward to implement and can be seamlessly integrated into the existing point cloud analysis pipeline.

Our contributions are summarized as follows:

- We first comprehensively revisit the long-existing sample protocol for point cloud learning through the lens of data corruption. Based on the analysis, we propose an alternative protocol to enhance the robustness of point cloud learning.
- We develop three learning-free techniques as the key of protocol, point reweighting, local-global balanced sampling, and local-geometry-preserved interpolation which can deal with point cloud corruption in different aspects. The whole proposed protocol is free of model architecture change and extra learning, thus it can be implemented to replace the current protocol with minimal pains and fits almost all 3D deep models.
- Extensive experiments are conducted on synthesis and real corrupted 3D point cloud datasets. The results have demonstrated that the proposed protocol is able to improve the robustness of 3D point cloud classification and outperform the latest methods.

2 Related Work

Point Cloud Sampling. Point cloud sampling techniques typically consist of: 1) downsampling, also known as "simplification" [Dovrat *et al.*, 2019], and 2) upsampling [Zhang *et al.*, 2022b]. These techniques are divided into non-learning and learning-based methods [Zhang *et al.*, 2022b]. Traditional non-learning-based downsampling techniques include Farthest Point Sampling (FPS) [Eldar *et al.*, 1997], Random Sampling (RS) [Hu *et al.*, 2020], Poisson Disk Sampling (PDS) [Ying *et al.*, 2013], and voxelization [Lv *et al.*, 2021]. Conversely, learning-based downsampling methods account for downstream tasks [Dovrat *et al.*, 2019; Lang *et al.*, 2020; Qian *et al.*, 2020; Qian *et al.*, 2023]. Upsampling is categorized into learning-based [Yu *et al.*, 2018; Qiu *et al.*, 2022; He *et al.*, 2023a; He *et al.*, 2023b] and non-learning-based approaches [Alexa *et al.*, 2003; Huang *et al.*, 2013; Wu *et al.*, 2015a]. The non-learning-based sampling techniques are particularly susceptible to outliers due to their inherent structural limitations; meanwhile, the learning-based methods are either also sensitive to noise or dependent on downstream tasks and prone to overfitting. This paper introduces simple yet effective point cloud sampling techniques to overcome these challenges.

Point Cloud Classification. PointNet [Qi *et al.*, 2017a] has been a trailblazer in utilizing deep learning for point cloud analysis, with notable extensions such as PointNet++ [Qi *et al.*, 2017b], GDANet [Xu *et al.*, 2021], Point Transformer (PCT) [Guo *et al.*, 2021], and CurveNet [Xiang *et al.*, 2021]. However, the performance of these models significantly deteriorates with corrupted real-world data [Uy *et al.*, 2019; Ren *et al.*, 2022; Sun *et al.*, 2022]. To tackle this issue, existing literature offers three main types of solutions. The first focuses on modifying the model by altering its structure or training strategies, such as pooling operations based on sorting [Sun *et al.*, 2020] and model aggregation [Dong *et al.*, 2020]. The second type includes certified methods, exemplified by Pointguard, which theoretically enhances model robustness through certified classification [Liu *et al.*, 2021]. The third type is data-driven approaches that directly cleanse corrupted data, with notable methods including IF-defense [Wu *et al.*, 2020] and DUP-Net [Zhou *et al.*, 2019]. This paper aims to advance robustness from a new perspective by refining point cloud sampling protocol during data preparation, while the non-trivial architecture modification is avoided.

Point Cloud Data Augmentation. Point cloud augmentation is a widely recognized practice in the deep learning community, employed to improve the generalization capabilities of neural networks. Traditional augmentation methods, including random scaling, rotation, and jitter, are somewhat limited in their effectiveness for point cloud analysis [Zhu *et al.*, 2024]. Recent advancements have introduced sophisticated techniques such as PointCutMix [Zhang *et al.*, 2022a], PointAugment [Li *et al.*, 2020], PointMixup [Chen *et al.*, 2020], and PointWOLF [Kim *et al.*, 2021]. However, they suffer from various limitations. For instance, while PointMixup [Chen *et al.*, 2020] and PointWOLF [Kim *et al.*, 2021] largely rely on predefined transformations, PointAugment [Li *et al.*, 2020] emphasizes global transformations, often at the expense of local geometric details. To our knowledge, the sampling augmentation of point cloud for robust classification has been largely unexplored. In this work, we aim to enhance sampling protocols specifically tailored for robust point cloud classification, addressing this critical gap.

3 Proposed Sampling Protocol

3.1 Existing Sampling Protocol

Mainstream 3D point cloud classification frameworks employ a standardized protocol predicated on processing clean point clouds of predetermined size, e.g., 1,024. Formally, the input point cloud is defined as $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}$, where each point $\mathbf{p}_i \in \mathbb{R}^3$ and N represents a fixed size. A common technique of this protocol is the Farthest Point Sampling (FPS) algorithm, which identifies key points that serve as anchors for subsequent feature aggregation or pooling operations as in [Qi *et al.*, 2017b]. FPS iteratively constructs a subset $\mathcal{S} \subseteq \mathcal{P}$ by selecting points $\mathbf{s}_t$ according to:

$$\mathbf{s}_t = \arg\max_{\mathbf{p}_i \in \mathcal{P}} \min_{\mathbf{s} \in \mathcal{S}} \|\mathbf{p}_i - \mathbf{s}\|_2, \tag{1}$$

where $t = 1, \cdots, |\mathcal{S}|$ denotes the iteration index.

This conventional protocol, however, exhibits significant limitations when confronted with real-world scenarios where point clouds are invariably corrupted and of variable size. We can characterize a corrupted point cloud $\mathcal{P}'$ through the following formulation:

$$\mathcal{P}' = \mathcal{P} \backslash \mathcal{P}_s \cup \mathcal{O}, \tag{2}$$

where $\mathcal{P}_s$ represents points removed from the original clean cloud $\mathcal{P}$, while $\mathcal{O}$ comprises introduced noise points or outliers, typically arising from occlusion effects and sensor imperfections. This formulation exposes three critical vulnerabilities in the current protocol:

- FPS inherently selects outliers ($\mathcal{O}$) as key points due to its distance-based criterion.
- Missing points ($\mathcal{P}_s$) lead to information loss without compensatory mechanisms.
- The variable size of $\mathcal{P}_s$ and $\mathcal{O}$ violate the fixed-size input constraint.

Recent empirical studies [Ren *et al.*, 2022; Sun *et al.*, 2022] have confirmed these shortcomings, showing substantial performance degradation under corruption scenarios. This necessitates a fundamental redesign of the sampling protocol to enhance robustness in point cloud processing.

3.2 New Sampling Protocol

We propose the enhanced and learning-free sampling protocol towards **key points selection** and **full points preprocessing**, respectively. The new point cloud sampling protocol (PointSP) is visualized in Figure 2.

Key Points Sampling. We propose distinct sampling strategies for training and inference stages to select key points. During training, we employ a stochastic sampling approach based on point-wise weights by $\mathbf{s}_t \sim \text{Cat}(\mathcal{P}; \mathbf{w})$, where $\text{Cat}(\cdot;\cdot)$ denotes the categorical distribution, and $\mathbf{w} = \{w_1, \cdots, w_N\}$ represents the point weights. These weights are derived from the isolation rate of each point (detailed in Section 3.3), with more isolated points receiving lower weights to reduce their sampling probability. During inference, we introduce Filtered FPS (FFPS), a deterministic sampling method that modifies the conventional FPS by incorporating binary weights:

$$\mathbf{s}_t = \arg\max_{\mathbf{p}_i \in \mathcal{P}} \hat{w}_i \min_{\mathbf{s} \in \mathcal{S}_t} \|\mathbf{p}_i - \mathbf{s}\|_2. \tag{3}$$

where $\hat{w}_i$ is a binary weight determined by thresholding point weights at the ω-th quantile (typically $\omega = 0.95$) of all weights. This effectively filters out the most isolated points, which are often outliers or noise. The binarization simplifies implementation while maintaining effectiveness. Detailed weighting calculation is presented in Section 3.3.

Full Points Resampling. Our resampling protocol differentiates training and inference stages. During training, we deliberately introduce size variations to enhance model robustness against real-world point clouds often with non-uniform sizes. Given an input point cloud $\mathcal{P}$ with N points, we randomly adjust its size to $N + \Delta N$, where the random variable ΔN can be either positive (upsampling) or negative (downsampling). The modified point cloud $\tilde{\mathcal{P}}$ is obtained through:

$$\tilde{\mathcal{P}} = \begin{cases} \mathcal{P} \cup \Delta\mathcal{P}, & \text{if } \Delta N > 0 \\ \mathcal{P} \setminus \Delta\mathcal{P}, & \text{if } \Delta N \leqslant 0 \end{cases} \tag{4}$$

During inference, our protocol ensures consistent point cloud size by applying conditional upsampling. Specifically, for any input point cloud $\mathcal{P}$ with insufficient points (i.e., $|\mathcal{P}| < N$), we supplement it with additional points via upsampling to reach the target size N, namely, $\tilde{\mathcal{P}} = \mathcal{P} \cup \Delta\mathcal{P}$. The specific techniques for generating additional points ($\Delta\mathcal{P}$) in upsampling and downsampling are detailed in Section 3.3.

3.3 Proposed Sampling Techniques

We introduce three key techniques of the new sampling protocol, namely, designed downsampling and upsampling techinques for full points resampling, and the point reweighting techinque for key point sampling.

Point Reweighting. The point-wise weight in key points sampling can be defined by the concept of *Isolation Rate*. At first, we calculate the radius of a sphere containing k nearest neighbors of each point in $\mathcal{P}$, which is given by $r_i = \max_{\mathbf{q}_j \in \mathcal{N}_i^k} \|\mathbf{p}_i - \mathbf{q}_j\|_2$, where $\mathcal{N}_i^k \subseteq \mathcal{P}$ is the set of k neighbors of i-th point $\mathbf{p}_i$. We further define *Isolation Rate* for each point as w'_i, given by,

$$w'_i = \Pr_{d \in \mathcal{D}_i}(d \geqslant \bar{r}), \mathcal{D}_i = \{\|\mathbf{q}_j - \mathbf{p}_i\|_2 : \forall \mathbf{q}_j \in \mathcal{N}_i^k\} \tag{5}$$

where $\bar{r} = \text{Median}(\{r_i\}_{i=1}^N)$ is the median of all radius and $\Pr_Y(X)$ is the probability of X given condition Y. The isolation rate of a point suggests the extent of a point being isolated, *i.e.,* far from others in a probability way. Although a few associated concepts were proposed to calculate the exact local radius of points [Sotoodeh, 2006] and identify outliers, the isolation rate is naturally fit for point weighting due to the probability representation. In our key points sampling, we implement $w_i = 1 - w'_i$ and $\hat{w}_i = \mathbf{1}(w'_i \leqslant \omega)$ for training and inference, respectively, where $\mathbf{1}$ is the indicator function.

Downsampling: Local-global-balanced downsampling. For $\Delta N \leqslant 0$ in (4), we remove points through a flexible neighborhood-based approach. Specifically, we randomly select a center point $\mathbf{p}_i$ and consider its $\tilde{k}$ nearest neighbors $\mathcal{N}_i^{\tilde{k}}$, where $\tilde{k}$ is randomly sampled from $[|\Delta N|, N]$. The points to be removed are selected according to:

$$\Delta\mathcal{P} \sim \{\Gamma : \Gamma \subseteq \mathcal{N}_i^{\tilde{k}}, |\Gamma| = -\Delta N\} \tag{6}$$

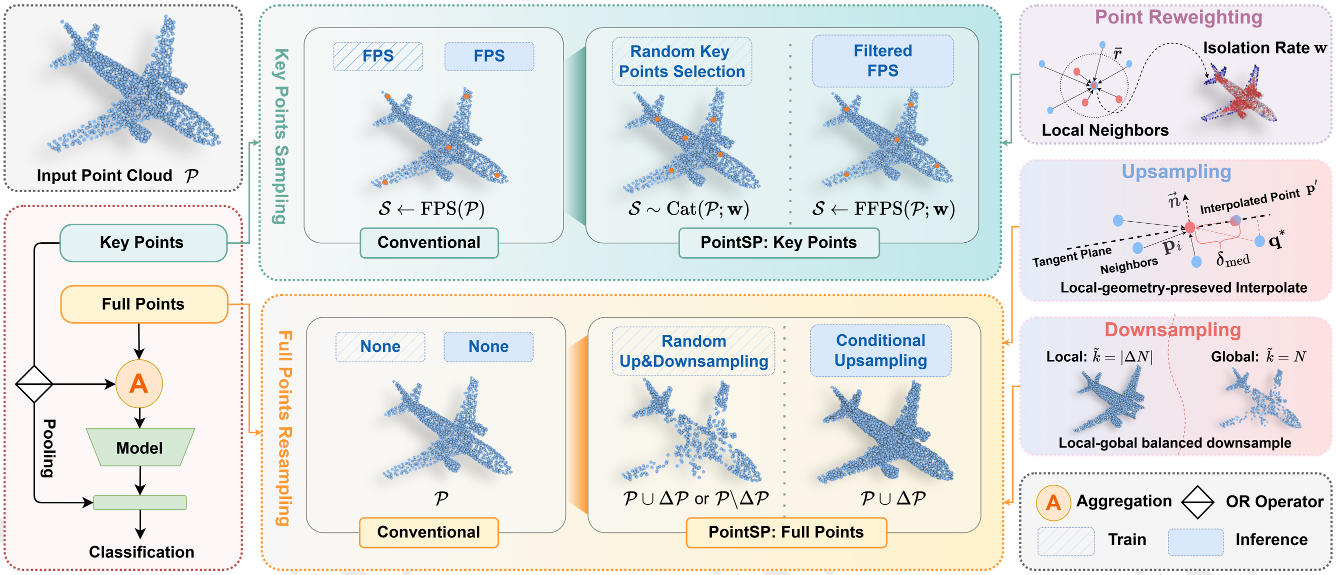
Input Point Cloud $\mathcal{P}$
Key Points
Full Points
A
Model
Pooling
Classification
Key Points Sampling
FPS
FPS
$\mathcal{S} \leftarrow \mathrm{FPS}(\mathcal{P})$
Conventional
Random Key Points Selection
Filtered FPS
$\mathcal{S} \sim \mathrm{Cat}(\mathcal{P}; \mathbf{w})$
$\mathcal{S} \leftarrow \mathrm{FFPS}(\mathcal{P}; \mathbf{w})$
PointSP: Key Points
Full Points Resampling
None
None
$\mathcal{P}$
Conventional
Random Up&Downsampling
Conditional Upsampling
$\mathcal{P} \cup \Delta\mathcal{P}$ or $\mathcal{P} \backslash \Delta\mathcal{P}$
$\mathcal{P} \cup \Delta\mathcal{P}$
PointSP: Full Points
Point Reweighting
$\bar{r}$
Isolation Rate w
Local Neighbors
Upsampling
$\vec{n}$
Interpolated Point $\mathbf{p}'$
Tangent Plane
Neighbors
$\mathbf{p}_i$
$\mathbf{q}^*$
δ_{med}
Local-geometry-preseved Interpolate
Downsampling
Local: $\tilde{k} = |\Delta N|$
Global: $\tilde{k} = N$
Local-gobal balanced downsample
A Aggregation
OR Operator
Train
Inference

Figure 2: PointSP: enhanced protocol of point cloud sampling for robust classification. The existing and conventional protocol used farthest point sampling (FPS) and non-processed points for input. In our protocol, randomized key point sampling and full points resampling (random up&downsampling) are used in training to conduct sampling-based data augmentation. During inference, filtered FPS (FFPS) is implemented to bypass outliers, and an upsampling strategy is used to densify sparse input. We propose the concept of isolation rate, the upsampling by tangent plane interpolation and the local-global balanced downsampling to obtain point weights and resampled points, respectively.

This strategy offers a continuous spectrum between local and global downsampling: when $\tilde{k} = |\Delta N|$, it removes a concentrated local patch; when $\tilde{k} = N$, it performs global random sampling; and intermediate values of $\tilde{k}$ provide balanced local-global downsampling. This flexibility helps simulate various real-world point cloud corruptions.

Upsampling: Local-geometry-preserved Interpolation. For $\Delta N > 0$, we propose a interpolation method that generates new points. For each point, we perform interpolation between it and a randomly selected neighbor on their shared tangent plane, as detailed in Algorithm 1. This approach ensures the preservation of local geometric features, particularly surface normals, while increasing point density. The upsampled points are sampled from the interpolation set $\hat{\mathcal{P}}$ by

$$\Delta\mathcal{P} \sim \{\Gamma : \Gamma \subseteq \hat{\mathcal{P}}, |\Gamma| = \Delta N\}. \tag{7}$$

This interpolation-based strategy effectively balances point cloud densification with geometric preservation, resulting in more natural and structurally coherent augmented data.

Computational Cost. We design the sampling protocol in the way that poses minimal computational effort beyond the original protocol. Particularly, the implementation of FPS in conventional protocol involves the calculation of point paired distances with complexity of $O(N^2)$. The proposed point reweighting and interpolation (for all points) techinques can utilize the same paired distances and induce extra operations with $O(kN)$ complexity. Such extra computation effort is minor since $k \ll N$.

Algorithm 1 Local Geometry Preserved Interpolation

```
Require: Point cloud $\mathcal{P}$, Point cloud normal $\{\mathbf{n}_i\}_{i=1}^N$ , A
    query point $\mathbf{p}_i \in \mathcal{P}$, Integer $k$
Ensure: A new point $\hat{\mathbf{p}}_i$
  $\mathcal{N}^k \leftarrow k\mathrm{NN}(\mathcal{P}, \mathbf{p}_i, k)$        ▷ $k$ nearest neighbors of $\mathbf{p}_i$
  $\delta_{\mathrm{med}} \leftarrow \mathrm{Median}(\{\|\mathbf{p}_i - \mathbf{q}\| : \mathbf{q} \in \mathcal{N}^k\})$  ▷ Compute the
    median of local distances as interpolation norm.
  $\mathbf{q}^* \sim \mathcal{N}^k$        ▷ Sample a random neighbor
  $\mathbf{v}_i \leftarrow (\mathbf{I} - \mathbf{n}_i\mathbf{n}_i^\top) \cdot (\mathbf{q}^* - \mathbf{p}_i)$   ▷ Compute the
    interpolation direction on tangent plane.
  $\hat{\mathbf{p}}_i \leftarrow \mathbf{p}_i + \delta_{\mathrm{med}} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$        ▷ Create interpolated point
  return $\hat{\mathbf{p}}_i$.
```

4 Experimental Studies

4.1 Experimental Setup

Dataset and Model. We utilize models trained on ModelNet40 [Wu *et al.*, 2015b] to conduct experiments on three corrupted datasets: ModelNet40-C, PointCloud-C, and OmniObject-C. The ModelNet40-C [Sun *et al.*, 2022] and PointCloud-C [Ren *et al.*, 2022] are datasets applying 15 and 7 distinct corruptions to ModelNet40's test set, totaling 2,468 objects. The OmniObject-C, based on OmniObject3D [Wu *et al.*, 2023], has real-scanned 362 objects corrupted by the methods proposed in [Ren *et al.*, 2022]. For 3D deep models, we employ PointNet [Qi *et al.*, 2017a], PointNet++ [Qi *et al.*, 2017b], GDANet [Xu *et al.*, 2021], CurveNet [Xiang *et al.*, 2021], PCT [Guo *et al.*, 2021], following the pipeline in ModelNet40-C including batch size and training protocol. We note that all experiments are run on NVIDIA GeForce

RTX 3090 GPUs.

Parameters Setting. The number of nearest neighbors k used in point weight computation is set to 20, which follows the common setup. During the inference phase, the key points sampling applies a threshold ω of 0.95, exploring the learning as depicted in Figure 4, meaning that FFPS filters out points within the lowest 5% of point weights.

Evaluation Protocol. We report the error rates (ER) and mean error rates (mER) across multiple corruptions on the three corrupted datasets for performance evaluation. A smaller ER indicates a superior performance.

4.2 Main Results

Overall Results. Mean error rates (mERs) for the three corrupted datasets are presented in Table 1. To facilitate a comprehensive comparison, we include multiple baseline models. The results clearly indicate that the proposed PointSP significantly enhances PCT and CurveNet; mERs decrease by approximately 10% across all datasets, with the most substantial improvement observed in PointCloud-C.

Results on ModelNet40-C. Extensive evaluations of PointSP on ModelNet40-C utilizing five 3D deep models revealed its superiority. Compared to five enhancement techniques (CutMix-R [Zhang *et al.*, 2022a], CutMix-K [Zhang *et al.*, 2022a], Mixup [Chen *et al.*, 2020], Rsmix [Lee *et al.*, 2021], and PGD [Sun *et al.*, 2021]), PointSP significantly improved all models. Across multiple corruption types, PointSP consistently achieved the lowest error rates: 24.1% for "*Density*", 9.5% for "*Noise*", and 11.1% for "*Transform*", demonstrating robustness. Notably, its unique randomized size sampling in resampling and FFPS in downsampling effectively tackled "*Density*" and "*Noise*" corruptions, enhancing resilience and eliminating outliers, respectively.

Results on PointCloud-C and OmniObject-C. Table 3 compares the performance of PointSP with data augmentation methods on the PointCloud-C and OmniObject-C datasets. On PointCloud-C, PointSP enhances classification accuracy across various corruption scenarios, outperforming other methods, with the FFPS technique achieving an error rate of 7.5% under additive corruption types due to its effective outlier filtering. However, PointSP does not perform optimally under the "*Jitter*" corruption, where the PGD strategy excels due to its robust feature learning mechanism. For drop-type corruptions, methods like CutMix and Rsmix demonstrate superior robustness, likely due to their data-mixing strategies. On OmniObject-C, PointSP excels in improving out-of-distribution (OOD) robustness, achieving the lowest mER for CurveNet (57.9%) and the best results for "*Jitter*" (61.4% mER). It also outperforms other methods on PointNet++ for "*Drop-G*" and "*Add-G*" corruptions, and is highly competitive with CutMix-R on PCT. Overall, these results validate PointSP's effectiveness in enhancing both OOD robustness and generalization.

Method	MNC	PCC	OmniC
PointNet [Qi *et al.*, 2017a]	28.3	33.7	65.2
PointNet++ [Qi *et al.*, 2017b]	30.6	27.7	73.9
DGCNN [Wang *et al.*, 2019]	25.9	23.5	73.7
RSCNN [Liu *et al.*, 2019]	26.2	26.1	72.4
CurveNet [Xiang *et al.*, 2021]	23.1	24.4	67.9
SimpleView [Goyal *et al.*, 2021]	27.2	24.3	71.8
GDANet [Xu *et al.*, 2021]	23.5	24.6	70.9
PCT [Guo *et al.*, 2021]	25.5	25.8	69.8
PCT+**PointSP**	**15.8**	**12.5**	60.8
CurveNet+**PointSP**	**15.8**	13.7	**57.9**

Table 1: Mean error rate (mER) across all corruptions of popular 3D deep models w/o our protocol on three datasets. The best mERs are highlighted in bold. MNC, PCC and OmniC represent ModelNet40-C, PointCloud-C and OmniObject-C, respectively.

Method	mER	Density	Noise	Transform
PointNet	28.3	28.3	32.7	24.0
+ CutMix-R	21.8	30.5	18.0	16.9
+ CutMix-K	21.6	26.8	21.8	16.3
+ Mixup	25.4	28.3	28.9	19.0
+ Rsmix	22.5	24.8	27.3	15.5
+ PGD	25.9	28.8	28.4	20.5
+ **PointSP**	21.6	26.0	18.6	20.2
PointNet++	30.6	36.9	30.3	24.6
+ CutMix-R	19.8	26.8	14.0	18.6
+ CutMix-K	21.3	24.9	19.3	19.6
+ Mixup	18.6	29.7	12.6	13.5
+ Rsmix	27.0	28.9	23.8	28.3
+ **PointSP**	17.7	**24.1**	12.7	16.2
GDANet	23.5	33.2	23.7	13.7
+ CutMix-R	16.9	28.5	10.4	11.9
+ CutMix-K	17.8	28.8	12.6	11.9
+ Mixup	18.5	30.3	13.1	12.2
+ Rsmix	19.2	27.7	14.4	15.4
+ PGD	20.3	32.1	15.9	13.0
+ **PointSP**	16.8	28.3	10.0	12.1
CurveNet	23.1	31.4	26.5	11.4
+ CutMix-R	16.1	25.7	10.5	12.1
+ CutMix-K	17.1	24.8	13.6	12.9
+ Mixup	20.8	32.4	17.9	12.1
+ Rsmix	19.9	26.7	15.6	17.3
+ PGD	20.4	28.5	21.3	11.4
+ **PointSP**	**15.8**	25.6	10.2	11.7
PCT	25.5	34.8	28.1	13.5
+ CutMix-R	16.3	27.1	10.5	11.2
+ CutMix-K	16.5	25.8	12.6	**11.1**
+ Mixup	19.5	30.3	16.7	11.5
+ Rsmix	17.3	25.0	12.0	15.0
+ PGD	18.4	29.3	14.7	**11.1**
+ **PointSP**	**15.8**	26.9	**9.5**	**11.1**

Table 2: Comparison of mean error rate (mER) on ModelNet40-C between PointSP and state-of-the-art point cloud augmentation methods across five 3D deep models. The best mERs for each 3D deep model are underlined.

4.3 Results for Part Segmentation

The proposed sampling protocol has been evaluated on a classification task. To demonstrate its broader applicability, we also applied it to part segmentation tasks, which are critical for robotic manipulation, using the ShapeNet-C dataset [Ren *et al.*, 2022]. The results, shown in Figure 3, clearly indi-

Method	PointCloud-C								OmniObject-C							
	mER	Scale	Jitter	Drop-G	Drop-L	Add-G	Add-L	Rotate	**mER**	Scale	Jitter	Drop-G	Drop-L	Add-G	Add-L	Rotate
PointNet++	27.7	9.4	50.3	26.2	39.6	15.9	20.2	32.5	73.9	64.9	80.7	77.9	78.5	71.6	71.1	72.4
+ CutMix-R	17.5	8.8	34.5	**9.0**	20.9	7.7	8.1	33.7	64.9	62.9	72.2	62.8	67.2	59.3	58.1	71.8
+ CutMix-K	19.1	9.2	45.0	12.8	16.0	8.1	9.5	33.4	64.8	60.9	76.1	65.6	64.6	57.7	60.1	68.8
+ Mixup	17.7	**8.5**	25.2	16.4	27.2	9.5	11.7	25.4	62.8	59.3	67.1	64.1	69.2	57.8	57.2	65.3
+ Rsmix	21.3	9.9	54.3	12.0	14.3	7.9	8.9	41.9	66.7	63.8	78.1	66.2	68.1	59.4	60.6	70.6
+ **PointSP**	17.3	9.2	33.3	10.9	16.0	8.0	10.3	33.1	62.4	59.9	72.0	58.6	64.1	56.8	57.8	67.7
CurveNet	24.4	8.9	22.9	17.3	22.3	52.1	28.7	18.9	67.9	59.4	67.7	63.0	68.7	80.4	70.4	65.9
+ CutMix-R	13.8	9.1	18.2	11.1	15.5	8.1	11.0	22.4	63.2	61.0	68.2	59.8	65.4	58.6	61.7	68.0
+ CutMix-K	15.8	8.7	30.6	12.9	10.3	8.5	15.5	23.9	60.3	**57.3**	69.1	59.7	**56.0**	**53.2**	63.4	**63.4**
+ Mixup	19.3	8.6	17.9	21.6	19.8	25.7	20.1	21.4	62.5	57.7	62.7	60.9	65.7	64.1	61.7	64.6
+ Rsmix	16.9	9.1	35.0	11.0	10.2	9.2	13.1	30.9	62.2	59.2	73.5	57.4	59.5	56.5	61.9	67.8
+ PGD	22.7	16.8	11.2	12.9	26.0	48.9	25.3	18.3	67.4	68.9	62.8	59.2	69.1	75.5	67.2	69.3
+ **PointSP**	13.7	10.3	19.0	10.3	11.0	7.6	15.5	22.0	57.9	58.1	**61.4**	**55.0**	**56.0**	54.1	**56.6**	64.0
PCT	25.8	9.0	27.1	15.0	24.1	40.3	42.9	22.2	69.8	59.3	71.3	60.4	68.7	83.0	80.6	65.5
+ CutMix-R	12.7	10.1	14.5	9.8	14.3	8.3	10.9	20.7	60.8	59.5	62.9	58.8	60.8	57.7	61.2	64.7
+ CutMix-K	14.1	9.5	22.3	11.3	10.2	8.5	15.6	21.2	61.4	**57.3**	65.8	62.8	58.8	56.3	65.1	63.5
+ Mixup	18.1	9.4	15.6	15.8	18.2	23.5	22.8	21.1	62.7	57.6	62.0	58.9	63.6	65.7	67.5	63.5
+ Rsmix	15.2	9.3	25.7	10.2	**10.0**	8.7	13.0	29.8	63.3	59.2	70.5	60.6	59.8	58.6	65.9	68.5
+ PGD	20.0	14.6	**10.5**	16.9	24.8	29.5	22.7	21.2	65.6	65.7	61.5	66.7	71.8	62.6	64.3	66.7
+WOLFMix	12.7	9.4	27.0	9.4	10.2	8.8	13.9	**10.5**	60.5	59.3	61.9	59.1	60.7	58.2	60.1	64.5
+ **PointSP**	**12.5**	9.9	16.2	11.0	14.4	**7.5**	**7.5**	21.1	60.8	60.5	65.1	58.1	60.9	57.7	57.5	66.0

Table 3: Comparison of mER on PointCloud-C and OmniObject-C datasets between PointSP and state-of-the-art point cloud enhancement methods across three 3D deep models.

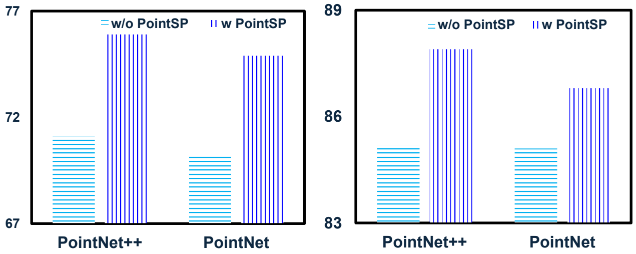

Figure 3: Instance mean IoU (left) and overall accuracy (right) on ShapeNet-C.

cate that the proposed PointSP protocol provides a significant improvement.

4.4 Ablation Studies

In the ablation studies, we use PCT [Guo *et al.*, 2021] and CurveNet [Xiang *et al.*, 2021] as 3D deep models on two datasets: ModelNet40-C [Sun *et al.*, 2022] and PointCloud-C (PCC) [Ren *et al.*, 2022].

Sampling Protocols in Training. We compared various sampling protocols during the training phase. As shown in Table 4, the combination of random up&downsampling (RUD) and stochastic weighted sampling (SWS) consistently delivers the best performance. Removing RUD or substituting the proposed SWS method with FPS obviously degrades the performance. FFPS also achieves the second-based performance, highlighting the importance of point reweighting.

Full Points Resampling	Key Points Sampling			PCT		CurveNet	
RUD	FPS	FFPS	**SWS**	MNC	PCC	MNC	PCC
			✓	19.2	14.8	22.7	20.1
✓	✓			16.8	13.3	16.3	14.1
✓		✓		17.0	13.3	16.0	13.8
✓			✓	**15.8**	**12.5**	**15.8**	**13.7**

Table 4: Ablation study based on mER with different sampling protocols **in training**. RUD: random upsampling & downsampling; FFPS: filtered FPS; SWS: stochastic weighted sampling.

Random Upsampling and Downsampling. We investigate the impact of various resampling techniques during training. As shown in Table 5, we explore different methods for increasing and reducing sample sizes. The results indicate that the proposed local-glocal-balanced method plays a crucial role in enhancing performance. This suggests that stochastically determining the localness of the point dropping, *i.e,* neighbor size, can improve robustness against corruptions. Additionally, we compare our upsampling method in Algorithm 1 with Shape-invariant perturbation (SI), which conducts per-point perturbation on the tangent plane [Huang *et al.*, 2022]. The superiority of our method over SI shows that perturbation direction based on neighbor points preserves more local information than random directions.

Effect of Quantile-based Threshold ω of FFPS. As illustrated in Figure 4, retain 95% of the points (i.e., filtering out the 5% of points with the smallest weights) results in the best performance. Moreover, when nearly half of the points are removed, the mER peaks, likely due to the loss of critical information within the point cloud. As the number of removed points decreases, the error rate also decreases, reaching its

$\Delta N > 0$		$\Delta N \leqslant 0$			PCT		CurveNet	
SI	**LGP**	RD	KNN	**LGB**	MNC	PCC	MNC	PCC
✓				✓	**15.8**	12.6	17.2	14.7
	✓	✓			17.0	13.4	16.7	14.8
	✓		✓		17.2	13.8	17.4	15.9
	✓			✓	**15.8**	**12.5**	**15.8**	**13.7**

Table 5: Ablation study based on mER of **full points resampling** using different techniques for upsampling ($\Delta N > 0$) and removing points ($\Delta N \leqslant 0$). SI: shape-invariant tangent plane upsampling with random directions and distances; **LGP**: the proposed local-geometry-preserved upsampling; RD: random global downsampling of points; KNN: local neighbours removal; **LGB**: the proposed local-gobal balanced downsampling.

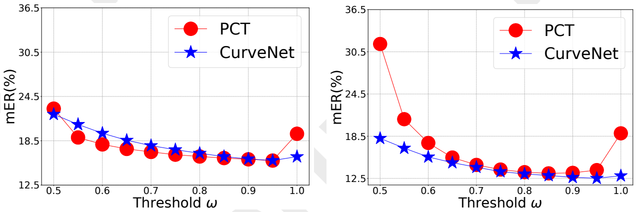

Figure 4: mERs of FFPS's different threshold ω on ModelNet40-C (left) and PointCloud-C (right).

lowest when about 5% of the points are filtered out, before rising again. It suggests that an optimal balance is achieved.

4.5 Visualization Results

Isolation Rate. In Figure 5, we visualize the distribution of point-wise isolation rates for three example objects. The proposed rate effectively identifies boundary points and outliers, thereby enhancing subsequent point cloud sampling and improving learning robustness against corruption.

Local-geometry-preserved Interpolation. Figure 6 visually compares the results of three upsampling techniques on four example objects. It is evident that both Jitter and SI [Huang *et al.*, 2022] struggle with corrupted data, particularly when it is sparse and non-uniform. In contrast, the proposed LGP method effectively combines completion and uniformity in the upsampling process.

Neighborhood Size $\tilde{k}$ in Local-global-balanced downsampling. Stochastically determining the sample size is a critical aspect of the resampling protocol. As shown in Figure 7, a smaller $\tilde{k}$ leads to local drops (second row), while a larger $\tilde{k}$ results in more global removals (last row). A stochastic $\tilde{k}$ would closely mimic real-world corruption, contributing to the robust improvement of the proposed protocol.

5 Conclusion

This work highlights the limitations of current sampling protocols for corrupted 3D point clouds and proposes PointSP, a robust solution that mitigates outliers and restores incomplete clouds without additional training. Experiments show that PointSP outperforms state-of-the-art methods in 3D classification, paving the way for more reliable 3D deep learning in critical applications like autonomous driving.

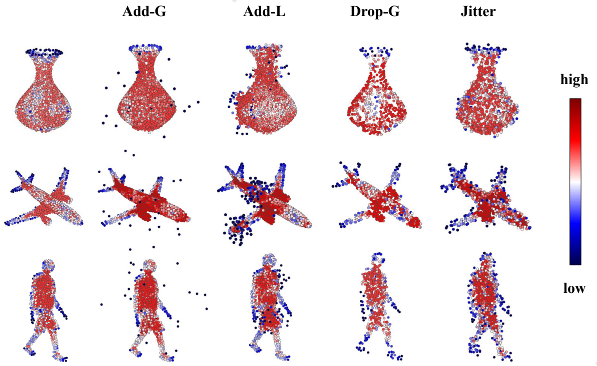

Figure 5: Visualization of point-wise *Isolation Rate*. Column 1 presents the clean data, while Columns 2 to 5 depict data with the corresponding corruption types indicated above each column.

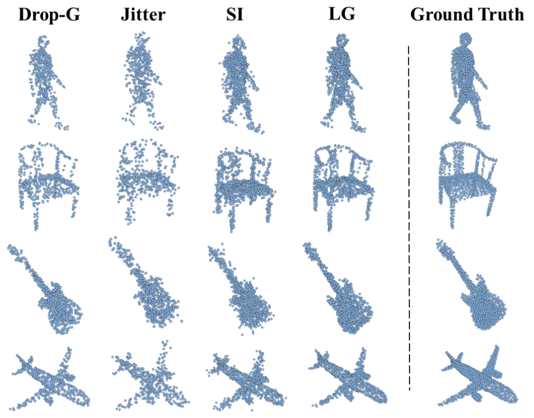

Figure 6: Visual comparison of three sampling techniques: (1) Jitter, (2) SI: tangent plane upsampling with random directions and distances and (3) LG: the proposed local-geometry-preserved upsampling. The first column shows the corrupted data from PointCloud-C, while the last column presents the corresponding clean data.

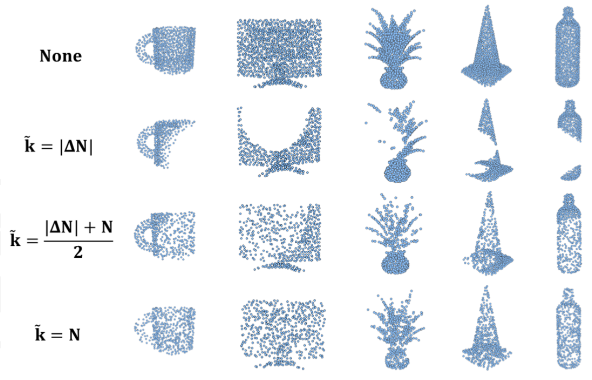

Figure 7: Visualization of samples with different neighborhood size $\tilde{k}$ in local-global-balanced downsampling.

Acknowledgments

This research was partially supported by National Natural Science Foundation of China (Grant No. 62202395 & 62176221), Sichuan Science and Technology Program (Grant No: 2024NSFTD0036 & 2024ZHCG0166), and the Fundamental Research Funds for the Central Universities (Grant No: 2682025ZTZD013).

References

[Alexa *et al.*, 2003] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graph.*, 9(1):3–15, 2003.

[Chen *et al.*, 2020] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *Eur. Conf. Comput. Vis.*, pages 330–345. Springer, 2020.

[Dong *et al.*, 2020] Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11513–11521. IEEE, 2020.

[Dovrat *et al.*, 2019] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2760–2769, 2019.

[Eldar *et al.*, 1997] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.*, 6(9):1305–1315, 1997.

[Goyal *et al.*, 2021] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *Proc. Int. Conf. Mach. Learn.*, pages 3809–3820. PMLR, 2021.

[Guo *et al.*, 2020] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4338–4364, 2020.

[Guo *et al.*, 2021] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Comput. Vis. Media (Beijing)*, 7:187–199, 2021.

[He *et al.*, 2023a] Yun He, Danhang Tang, Yinda Zhang, Xiangyang Xue, and Yanwei Fu. Grad-pu: Arbitrary-scale point cloud upsampling via gradient descent with learned distance functions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5354–5363, 2023.

[He *et al.*, 2023b] Yun He, Danhang Tang, Yinda Zhang, Xiangyang Xue, and Yanwei Fu. Grad-pu: Arbitrary-scale point cloud upsampling via gradient descent with learned distance functions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5354–5363, 2023.

[Hong *et al.*, 2023] Cheng-Yao Hong, Yu-Ying Chou, and Tyng-Luh Liu. Attention discriminant sampling for point clouds. In *Int. Conf. Comput. Vis.*, pages 14429–14440, 2023.

[Hu *et al.*, 2020] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11108–11117, 2020.

[Huang *et al.*, 2013] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM Trans. Graph.*, 32(1):1–12, 2013.

[Huang *et al.*, 2022] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3d adversarial point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15335–15344, 2022.

[Kim *et al.*, 2021] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J Kim. Point cloud augmentation with weighted local transformations. In *Int. Conf. Comput. Vis.*, pages 548–557, 2021.

[Lang *et al.*, 2020] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7578–7588, 2020.

[Lee *et al.*, 2021] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15900–15909, 2021.

[Li *et al.*, 2020] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Pointaugment: an auto-augmentation framework for point cloud classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6378–6387, 2020.

[Liu *et al.*, 2019] 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.

[Liu *et al.*, 2021] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Pointguard: Provably robust 3d point cloud classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6186–6195, 2021.

[Lv *et al.*, 2021] Chenlei Lv, Weisi Lin, and Baoquan Zhao. Approximate intrinsic voxel structure for point cloud simplification. *IEEE Trans. Image Process.*, 30:7241–7255, 2021.

[Qi *et al.*, 2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 652–660, 2017.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inform. Process. Syst.*, 30, 2017.

[Qian *et al.*, 2020] Yue Qian, Junhui Hou, Sam Kwong, and Ying He. Pugeo-net: A geometry-centric network for 3d point cloud upsampling. In *Eur. Conf. Comput. Vis.*, pages 752–769. Springer, 2020.

[Qian *et al.*, 2022] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Adv. Neural Inform. Process. Syst.*, 35:23192–23204, 2022.

[Qian *et al.*, 2023] Yue Qian, Junhui Hou, Qijian Zhang, Yiming Zeng, Sam Kwong, and Ying He. Task-oriented compact representation of 3d point clouds via a matrix optimization-driven network. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(11):6981–6995, 2023.

[Qiu *et al.*, 2022] Shi Qiu, Saeed Anwar, and Nick Barnes. Pu-transformer: Point cloud upsampling transformer. In *Asian Conf. Comput. Vis.*, pages 2475–2493, 2022.

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

[Sotoodeh, 2006] Soheil Sotoodeh. Outlier detection in laser scanner point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):297–302, 2006.

[Sun *et al.*, 2020] 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.

[Sun *et al.*, 2021] 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. *Adv. Neural Inform. Process. Syst.*, 34:15498–15512, 2021.

[Sun *et al.*, 2022] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z Morley Mao. Modelnet40-c: A robustness benchmark for 3d point cloud recognition under corruption. In *ICLR 2022 Workshop on Socially Responsible Machine Learning*, volume 7, 2022.

[Uy *et al.*, 2019] 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 *Int. Conf. Comput. Vis.*, pages 1588–1597, 2019.

[Wang *et al.*, 2019] 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 Trans. Graph.*, 38(5):1–12, 2019.

[Wu *et al.*, 2015a] Shihao Wu, Hui Huang, Minglun Gong, Matthias Zwicker, and Daniel Cohen-Or. Deep points consolidation. *ACM Trans. Graph.*, 34(6):1–13, 2015.

[Wu *et al.*, 2015b] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1912–1920, 2015.

[Wu *et al.*, 2020] 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.

[Wu *et al.*, 2023] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 803–814, 2023.

[Xiang *et al.*, 2021] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Int. Conf. Comput. Vis.*, pages 915–924, 2021.

[Xu *et al.*, 2021] 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 *Proc. AAAI Conf. Artif. Intell.*, volume 35, pages 3056–3064, 2021.

[Yan *et al.*, 2020] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5589–5598, 2020.

[Ying *et al.*, 2013] Xiang Ying, Shi-Qing Xin, Qian Sun, and Ying He. An intrinsic algorithm for parallel poisson disk sampling on arbitrary surfaces. *IEEE Trans. Vis. Comput. Graph.*, 19(9):1425–1437, 2013.

[Yu *et al.*, 2018] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2790–2799, 2018.

[Yu *et al.*, 2022] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 19313–19322, 2022.

[Zhang *et al.*, 2022a] 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.

[Zhang *et al.*, 2022b] Yan Zhang, Wenhan Zhao, Bo Sun, Ying Zhang, and Wen Wen. Point cloud upsampling algorithm: A systematic review. *Algorithms*, 15(4):124, 2022.

[Zhou *et al.*, 2019] 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 *Int. Conf. Comput. Vis.*, pages 1961–1970, 2019.

[Zhu *et al.*, 2024] Qinfeng Zhu, Lei Fan, and Ningxin Weng. Advancements in point cloud data augmentation for deep learning: A survey. *Pattern Recognit.*, page 110532, 2024.