This ACCV 2024 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2024
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

DA^2: Degree-Accumulated Data Augmentation on Point Clouds with Curriculum Dynamic Threshold Selection

Ta Chun Tai[1], Nhat-Tuong Do-Tran[1], Ngoc-Hoang-Lam Le[1], Yung-Hui Li[2], and Ching-Chun Huang[1]*

[1] National Yang Ming Chiao Tung University, Taiwan
{maxtai.10, tuongdotn.cs11, lengochoanglam.ee12, chingchun}@nycu.edu.tw
[2] Hon Hai Research Institute, Taiwan
yunghui.li@foxconn.com

Abstract. Conventional point cloud data augmentation methods typically employ offline transformations with predefined, randomly applied transformations. This randomness may lead to suboptimal training samples that are not suitable for the current training stage. Additionally, the predefined parameter range restricts the exploration space of augmentation, limiting the diversity of samples. This paper introduces Degree-Accumulated Data Augmentation (DA^2), a novel approach that accumulates augmentations to expand the exploration space beyond predefined limits. We utilize a teacher-guided auto-augmenter to prevent the generation of excessively distorted or unrecognizable samples. This method aims to generate challenging yet suitable samples, progressively increasing the difficulty to enhance the model's robustness. Additionally, according to a student model's ability, we propose Curriculum Dynamic Threshold Selection (CDTS) to filter overly challenging samples, allowing the model to start with high-quality objects and gradually handle more complex ones as model stability improves. Our experiments show that this framework significantly enhances accuracy across various 3D point cloud classifiers.

Keywords: Degree-Accumulate Data Augmentation · Point Cloud Classification · Curriculum Dynamic Thresholding · Teacher-Student Model.

1 Introduction

With the advancement in computer vision, applications in robotics and autonomous vehicles require more precise information. Point clouds, which offer substantial spatial geometric information, are being utilized more frequently across various applications. However, collecting point cloud data is challenging, resulting in datasets that are significantly smaller than those for images. For instance, as illustrated in Table 1, the widely used ImageNet dataset [8] comprises approximately 1.4 million images across 1,000 categories, whereas the

* Corresponding author.

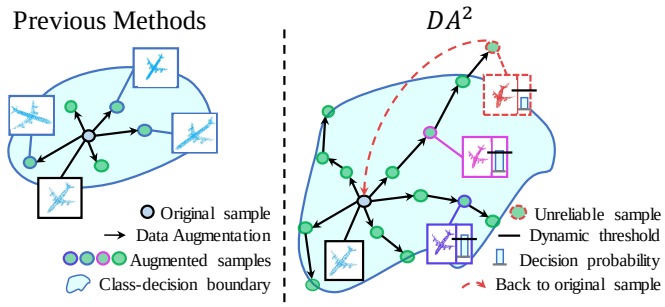

Fig. 1. Our proposed DA^2 method accumulates augmentations through iterations. This process significantly expands the augmentation space, and elevates the difficulty and diversity of data, thereby boosting the model's performance.

Table 1. The comparison of datasets shows the scarcity of 3D point cloud data in contrast to image datasets.

Dataset	Data Type	# of samples	# of classes
ImageNet [8]	Image	1.4M	1k
ModelNet40 [29]	Point Cloud	12K	40
ScanObjectNN [26]	Point Cloud	15K	15

ScanObjectNN point cloud dataset [26] contains only 15,000 samples across 15 categories. The limited size and diversity of point cloud datasets can hinder the generalization capabilities of 3D deep learning networks.

Data augmentation is a common method used to address this issue by artificially generating samples with different appearances to increase the size and diversity of datasets, allowing deep learning networks to learn more varied features. Common data augmentation methods for 3D point clouds include global rotation, scaling, translation, and transformations applied to the entire point cloud to generate diversity in viewpoints and sizes of the same object. Techniques such as jittering and adding noise disrupt the fine structure of point clouds, simulating data collected by sensors in the real world and making the model more robust. There are also methods like PointWOLF [12] that apply transformations to local regions of the point cloud to increase geometric diversity.

While these offline data augmentation methods are easy to use and effectively improve model generalization, they require predefined parameter ranges to prevent generating unrealistic objects and learning incorrect information. The choice of parameters significantly impacts model performance, and as the number of methods increases, finding suitable parameters becomes increasingly time-consuming. Additionally, during training, the model does not consider what samples it should be learning at any given time. If overly simple or difficult samples are generated, it can lead to overfitting or hinder the model's ability to learn, causing negative effects.

To address these challenges, we propose Degree-Accumulated Data Augmentation (DA^2), a novel online data augmentation training strategy. As illustrated in Fig. 1, by accumulating data augmentations, DA^2 breaks through the limitations of predefined parameter ranges, significantly increasing sample diversity. Tailored to various learning stages, DA^2 employs a trainable augmenter to generate challenging data augmentations automatically. Unlike traditional methods that apply static transformations offline to original data each epoch, DA^2 utilizes a Degree-Accumulated augmentation process to explore a broader search space. It inputs augmented samples into subsequent epochs, progressively widening the exploration scope with each iteration. To address the potential for generating unsuitable samples, we incorporate a teacher model [24] to guide the augmenter, ensuring the production of beneficial and accurate samples for the target model.

Additionally, we introduce Curriculum Dynamic Threshold Selection (CDTS) to further refine the selection of suitable training samples based on the student network's capabilities at each stage of training, a process we also refer to as "Student Selection". Beyond just generating samples deemed beneficial from the teacher's perspective, it is crucial to align these with the student's current learning capacity. While it is advantageous for the model to tackle challenging samples, those that are excessively difficult can be detrimental. To address this, we employ curriculum learning principles, designed to guide the model from simpler to more complex tasks gradually. Feedback from the student model controls this process, allowing us to dynamically filter out samples that are too challenging, thereby implementing a curriculum-based sample selection. This strategy ensures that the model incrementally takes on more difficult samples, optimizing learning efficiency and effectiveness.

In summary, our contributions can be summarized as follows:

- We develop an automated data augmentation technique that can generate diverse data, while ensuring the data is efficient for training. Moreover, it is adaptive to different samples, models, and datasets.
- We propose a novel Degree-Accumulated Data Augmentation that significantly enhances the variability of samples.
- We introduce a Teacher-guided auto-augmenter that ensures augmented samples are valuable for learning.
- We introduce Curriculum Dynamic Threshold Selection to select suitable samples for the current training stage. Additionally, it can dynamically adapt to different datasets or models.
- Our method can be applied to any point cloud classifier, effectively improving performance.

2 Related Works

2.1 Deep Learning Techniques for Point Cloud Data

In computer vision, CNNs have achieved great success in 2D images. Early approaches to deep learning on point clouds utilize these existing techniques,

such as those presented in [11, 20, 23], which project 3D point clouds onto the 2D plane from multiple views for feature extraction using 2D CNN. However, these methods only partially exploit the unique characteristics of 3D point clouds, as they discard spatial geometric information.

Alternatively, methods such as those proposed in [5, 16, 31] partition the 3D space into voxels and employ 3D CNNs for feature extraction. While 3D CNNs leverage the three-dimensional nature of the data, they come with increased computational costs. Quantizing and projecting point clouds as voxels or pixels loses the spatial information of point cloud objects. To address this issue, Qi *et al.* [19] proposed PointNet. It is designed to capture three key properties of point clouds: unorderedness, interaction among points, and invariance under transformations. PointNet achieves point-wise feature extraction solely through multilayer perceptrons (MLPs) and max pooling, preserving the structure of point clouds. However, global max pooling in PointNet prevents it from capturing local features.

To capture local properties, PointNet++ [21], building upon PointNet, progressively increases the receptive field using hierarchical sampling and grouping to extract features at various scales. Many studies [10, 17, 22] have further explored PointNet++, achieving state-of-the-art results. On the other hand, DGCNN [28] emphasizes the importance of point-to-point relationships. It introduces EdgeConv to establish relationships between points and their neighboring nodes in a graph-based manner. This approach enables the extraction of local geometric information from point clouds while maintaining permutation invariance, thus enhancing feature extraction capabilities by reinforcing point relationships. Inspired by the success of Transformers in image feature extraction, Zhao *et al.* [30] introduced PointTransformer and applied it to 3D point clouds to capture global and local features, currently achieving state-of-the-art performance in this domain.

Numerous studies have focused on designing networks for point cloud feature extraction. With advancements in point cloud deep learning, these networks have become increasingly robust in their feature extraction capabilities. However, the size of training datasets has not grown proportionately, limiting the ability of these novel networks to demonstrate their true potential. To address this, we aim to enhance the diversity and complexity of datasets through data augmentation. By challenging the learning capabilities of point cloud deep learning networks, we seek to improve their performance and robustness.

2.2 Data Augmentation Techniques for Point Clouds

Data augmentation is widely used in deep learning to enhance dataset diversity, thereby improving model robustness and generalization. Conventional point cloud data augmentation methods are inherited from those in the 2D field, including random scaling, rotation, and translation of entire point clouds. These techniques allow models to learn features from objects of varying sizes and viewpoints. Additionally, methods such as adding noise, jittering, dropout, and occlusion simulate real-world conditions encountered by sensors collecting point cloud data.

However, these conventional methods only partially leverage the unique properties of point clouds. To address this, PointWOLF [12] applies scaling, rotation, and translation to local regions, altering the local structure and shape of the point cloud and enhancing the diversity of geometric features. Inspired by the success of label mixing in images, several mixup-based point cloud data augmentation techniques have been proposed in recent studies, such as PointMixup [3], PointSwapMix [25], and RSMix [13]. These methods combine different objects to enable models to extract features from mixed categories, thereby improving feature extraction capabilities.

Without considering the model status at different training stages, the aforementioned methods are classified as offline data augmentation. In these methods, parameter ranges are predefined, and augmentations are applied randomly within these ranges. However, suitable augmentation parameters vary across different models and even different stages of training. Instead of applying augmentations randomly, AutoAugment [6] uses a reinforcement learning strategy to find the best augmentation policies for the current stage. Subsequently, advanced studies such as Fast AutoAugment [15], RandAugment [7], and PBA [9] were proposed to find the optimal transformations more efficiently. These methods, however, are designed for 2D images. Cheng et al. [4] introduced PPBA, extending this technique to the 3D domain. Although these methods effectively improve model performance, the generated augmentation policies are applied uniformly across all data rather than being tailored to individual samples.

In response, some research has introduced trainable augmenters that predict transformation parameters corresponding to the input data, resulting in more robust training and sample-wise auto-augmentation. For example, PointAugment [14] uses a network to predict point cloud rotation, scaling, and point-wise translation, producing augmented point clouds for training. AdaptPoint [27] employs a GAN-based training strategy to generate corrupted samples, thereby enhancing model robustness. However, the previous methods generate augmented data that is used only for the current epoch, limiting the augmenter's ability to explore a more expansive sample space. Differing from these approaches, we have designed a novel accumulated augmentation strategy that retains augmented samples and continuously stacks augmentations. This method surpasses the limitations of predefined ranges, producing more challenging samples and creating a more comprehensive augmented dataset.

3 Proposed Methods

3.1 Degree-Accumulated Data Augmentation with Model Training

Our method, Degree-Accumulated Data Augmentation (DA^2), is an automated data augmentation technique coupled with the model training process. Given an input point cloud $\mathcal{P} \in \mathbb{R}^{n\times 3}$, our objective is to enhance this point cloud to produce $\mathcal{P}' \in \mathbb{R}^{n\times 3}$, a degree-accumulated augmented version that is both challenging and valuable for learning at the current time step. From time to time, by preserving and continuously accumulating augmentations, DA^2 breaks free

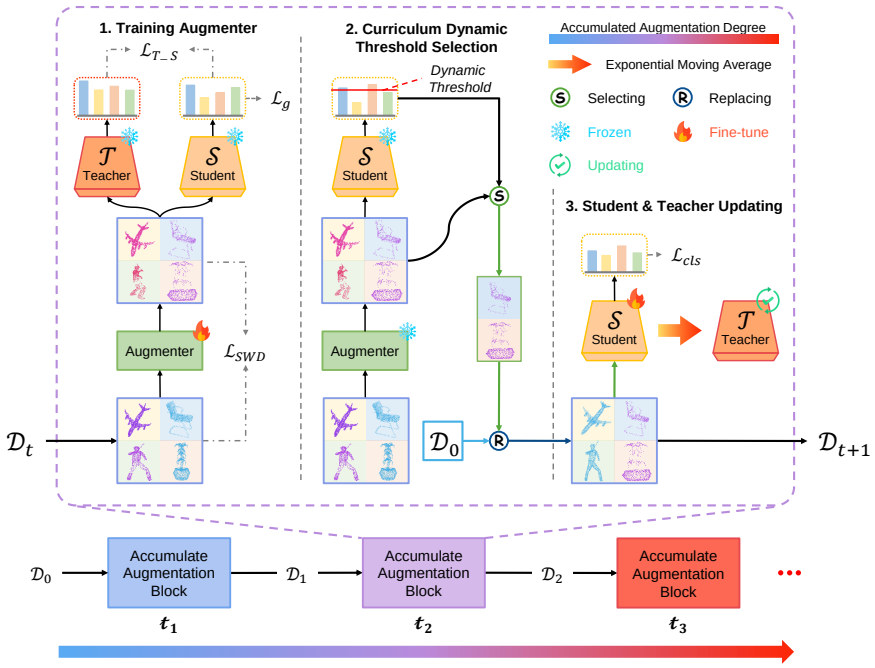

Fig. 2. Overall Architecture. At each time step, a corresponding AAB includes (a) a Training Augmenter phase, (b) a CDTS phase, and (c) a Teacher-Student model updating phase. First, we train the augmenter. Second, we use it to generate samples and filter them with CDTS to select suitable samples for training. Disqualified samples are replaced with their original form from D_0, forming D_{t+1}. Finally, we train the student and update the teacher using EMA. The color spectrum represents the accumulated augmentation degree, showing the variation of degree during the process.

from the constraints of predefined augmentation limits and significantly expands the range of augmented samples. Consequently, training with this augmented data enables more effective feature learning and enhances model robustness.

Specifically, we employ a Degree-Accumulated augmenter within a teacher-student training framework. Consequently, both the augmenter and the augmented samples are continuously updated during the training of the target network, namely the student network. The overall architecture, integrating DA^2 and model training, is depicted in Fig. 2. It includes several Accumulated Augmentation Blocks (AABs). At each time step, denoted as t_i, an AAB manages both data augmentation and model training, encompassing (a) a Training Augmenter phase, (b) a CDTS phase, and (c) a Teacher-Student model updating phase.

In the Training Augmenter phase, the frozen teacher and student models provide guidance to update the augmented samples. The trained augmenter aims to explore diverse and challenging augmentations for current input point clouds. During the CDTS phase, the updated augmenter generates adaptive and

diverse samples. However, at that specific training moment, some samples may be overly challenging for the student network. Using the student model's current capabilities as a reference, these complex samples are dynamically thresholded and reset to their original, unaltered form from the initial training dataset D_0, effectively preventing over-augmentation. Finally, in the Teacher-Student model updating phase, we retrain the student model and update the teacher model through exponential moving averaging of the student models.

3.2 Training Augmenter in AAB

Training Losses. We aim for the target model—a point-cloud classification model—to learn from sufficiently challenging samples, thereby efficiently driving its progress. Inspired by adversarial data augmentation techniques, we augment meaningful samples through adversarial learning. Traditional adversarial augmentation typically employs a GAN-based framework to generate samples, with the discriminator designed to differentiate between real and fake samples. However, this approach fails to consider the relationships between different classes.

To address this, we use a teacher-student framework to implement adversarial learning and train our augmenter. Both the teacher and student models are multi-class classifiers, where the teacher model accumulates long-term knowledge, and the student model focuses on short-term experiences. To enhance the student model's performance, we require adversarial samples that contain information not yet learned by the student but considered valuable by the teacher. Thus, we train our augmenter to generate samples that maximize the student model's classification error while minimizing the error for the teacher. This leads to the formulation of the teacher-student adversarial loss:

$$\mathcal{L}_{T\_S} = |1 - exp(CE(f_s(x'), y) - \lambda * CE(f_t(x'), y))|, \ x \in D_t \text{ and } |D_t| = N, \quad (1)$$

Here, $CE(\cdot)$ represents the cross-entropy loss. f_s and f_t are the student and teacher classifiers, respectively. λ is a hyperparameter that determines the influence of the teacher. x' denotes a sample augmented from its form at time t; that is $x \in D_t$. Finally, y represents the class label and N is the dataset size.

However, relying solely on the adversarial loss is insufficient. To prevent the augmented point clouds from deviating too far from their original form, we incorporate the Sliced Wasserstein distance (SWD) loss [2, 18]:

$$\mathcal{L}_{SWD} = SWD(\hat{x}, x'), \text{ where } \hat{x} \in D_0, \quad (2)$$

$$SWD(\hat{x}, x') = \frac{1}{P} \sum_{i=1}^{P} EMD\left(\mathcal{R}_{\theta_i} \hat{x}, \mathcal{R}_{\theta_i} x'\right). \quad (3)$$

In the equation, EMD measures the Earth Mover's Distance of two sets. $\mathcal{R}_{\theta_i}$ denotes a linear projection that projects 3D points onto a 2D plane θ_i, where $\theta_0, ..., \theta_P$ are sampled uniformly from the sphere $\mathbb{S}^2$. The SWD loss measures the

distance between the original point cloud $\hat{x}$ and the augmented point cloud x', ensuring the enhanced and original point clouds maintain similar distributions.

To enhance the augmenter's generative ability, a regularization loss is used:

$$\mathcal{L}_g = ||f_s(x')||^2. \tag{4}$$

This component encourages the augmenter to produce samples that are more likely to be confused with different classes, thus creating more challenging samples.

Finally, the total losses used to train the augmenter are as follows:

$$\mathcal{L}_{aug} = \mathcal{L}_{T\_S} + \mathcal{L}_{SWD} + \mathcal{L}_g. \tag{5}$$

Augmenter Architecture. To generate samples that incorporate global transformations and local structural deformations, we have developed a hierarchical augmentation process in three stages: global transformation, local deformation (utilizing PointWOLF [12]), and point-wise translation and masking. As depicted in Fig. 3, the process begins with an input point cloud $\mathcal{P}$. Initially, we apply global augmentations to $\mathcal{P}$, resulting in $\mathcal{P}_g$. These augmentations include scaling, rotation, and translation, which are applied randomly to preserve the intrinsic structure of the point cloud. For the more nuanced local and point-wise augmentations, we employ augmentation networks specifically designed to predict augmentation parameters, ultimately producing the final augmented version, $\mathcal{P}'$.

We leverage the ideas of PointWOLF [12] for point cloud deformation, which deforms point clouds by combining several local region transformations. Specifically, PointWOLF samples some points as anchors and then applies scaling, rotation, and translation to the point clouds centered on the anchor points. Finally, it combines these point clouds by interpolating them with kernel regression, resulting in a locally deformed point cloud.

In our design, our network predicts the deformation parameters for $\mathcal{P}_g$ to achieve auto augmentation. In this stage, the point cloud $\mathcal{P}g$ is fed into PointNet++ [21] to extract its features $\mathcal{F} \in \mathbb{R}^{n \times C}$. We then select several anchor points $\mathcal{A} \in \mathbb{R}^{m \times 3}$ within the point cloud using the Farthest Point Sampling (FPS). The features of the points surrounding each anchor are aggregated similarly to the PointNet Set Abstraction method, creating features for each anchor $\mathcal{F}_{anchor} \in \mathbb{R}^{m \times C'}$. These anchor features are then fed into MLPs to predict deformation parameters $\mathcal{D} = \{[\mathcal{S}_j, \mathcal{R}_j, \mathcal{T}_j] \mid j = 1, \ldots, m\} \in \mathbb{R}^{m \times d}$, where $\mathcal{S}_j$, $\mathcal{R}_j$, and $\mathcal{T}_j$ represent parameters of scaling, rotation, and translation for anchor j, respectively. The predicted parameters are used in the PointWOLF module to deform the point cloud, producing a deformed point cloud with diverse structures.

The final stage involves point-wise augmentation. Here, the point features, $\mathcal{F}$, are input into an MLP to generate point-wise translations, $\mathcal{T}^p = \{\mathcal{T}_i^p | i = 1, ..., n\} \in \mathbb{R}^{n \times 3}$, and point-wise masking, $\mathcal{M}^p = \{\mathcal{M}_i^p | i = 1, ..., n\} \in \mathbb{R}^{n \times 1}$. These translations and masks are then applied to the individual points, enabling fine-grained adjustments that refine the augmentation process. This stage ensures that even the smallest details within the point cloud are altered, thereby enhancing the output diversity and complexity. More details are provided in the Supplementary.

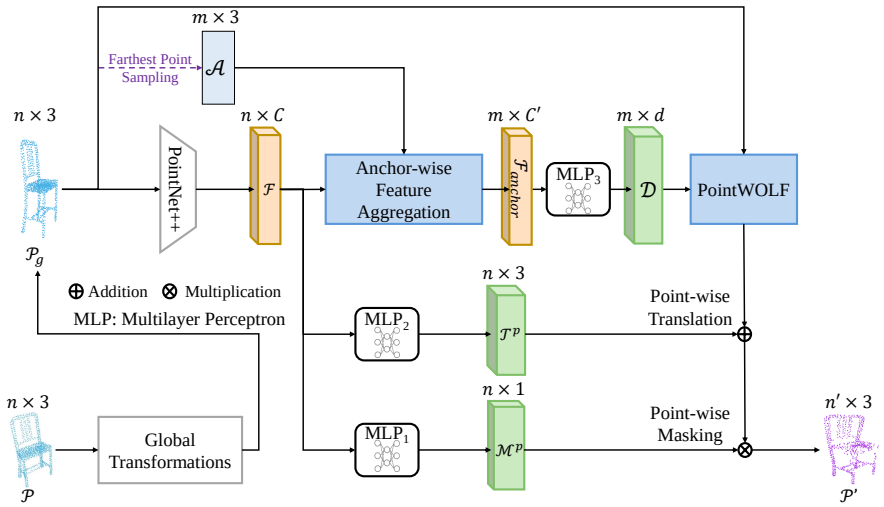

Fig. 3. Augmenter Architecture. The input point cloud $\mathcal{P}$ is first augmented with global transformations. Afterwards, the point cloud $\mathcal{P}_g$ is fed into PointNet++ to extract its features $\mathcal{F}$. Subsequently, we aggregate the features $\mathcal{F}$ to obtain anchor-wise features $\mathcal{F}_{anchor}$, where the anchors $\mathcal{A}$ are determined using FPS from $\mathcal{P}_g$. These anchor-wise features are then fed into an MLP to predict the deformation parameters $\mathcal{D}$ for PointWOLF [12]. Additionally, we use MLPs to predict point-wise masking $\mathcal{M}^p$ and translation $\mathcal{T}^p$. This process results in an output point cloud $\mathcal{P}'$, which is deformed, masked, and jittered.

3.3 Curriculum Dynamic Threshold Selection in AAB

Our method generates challenging and valuable samples, particularly through the accumulated augmentation strategy, which makes the samples more difficult than those produced by traditional augmentation methods. However, if the model lacks sufficient learning capability and cannot accurately extract features, this increased difficulty could be detrimental, leading to the learning of incorrect information. To address this, we propose CDTS, having two primary objectives. First, it prevents the student model from learning overly difficult samples. Second, because our DA^2 continually accumulates augmentations, CDTS provides a stopping mechanism. Samples whose classification scores fall below the threshold are filtered out and replaced with their original point clouds.

Curriculum. We also incorporate curriculum learning [1] techniques, allowing the model to progress from simple to complex samples. Initially, we establish a high classification score threshold, which allows the model to focus on higher-quality and less complex samples, thereby establishing a solid foundation. As training advances, the model's stability and learning capabilities improve, prompting a gradual decrease in the threshold. This adjustment progressively exposes the model to more challenging samples.

Dynamic. We observed that learning capabilities vary between different classes, with some harder-to-distinguish classes having significantly lower prediction scores than others. To address this issue, our threshold considers the average true positive prediction score of each class from previous iterations. By doing so, each class has a distinct threshold, allowing for more effective and precise filtering of suitable learning samples, thereby enhancing overall performance. Moreover, this dynamic adjustment mechanism ensures that our threshold selection can be directly applied to various datasets and different baseline classifiers. The equation of the Curriculum Dynamic Threshold is as follows:

$$Threshold\_ratio = \gamma^t * (r_h - r_l) + r_l, \tag{6}$$

$$Threshold_c = Threshold\_ratio \times \mu_c, \tag{7}$$

where r_h and r_l indicate the predefined upper and lower ratio, γ is the decay rate controlling how quickly the threshold decreases, t is the current training iteration, and μ_c is the average of true positive prediction scores for class c from the previous epoch. This equation adaptively lowers the threshold as training progresses, incorporating class-specific performance to find suitable samples.

3.4 Student and Teacher Model Updating in AAB

Finally, the augmented and filtered dataset is used for training. The student model is trained using a cross-entropy loss defined as follows:

$$\mathcal{L}_{cls} = CE(f_s(x), y),\ x \in D_{t+1}. \tag{8}$$

Due to our accumulated augmentation strategy, we aim for the teacher to retain past information. Therefore, the teacher classifier is updated by applying the weights of the student classifier using an Exponential Moving Average (EMA) method. Through this approach, the teacher preserves long-term memory, assisting the student in acquiring more knowledge.

4 Experimental Results and Discussions

4.1 Datasets and Evaluation Metrics

ScanObjectNN. ScanObjectNN [26] is a real-world point cloud object dataset derived from scanned indoor scene data, containing approximately 15,000 real scanned objects categorized into 15 classes. The dataset's real-world origin introduces challenges such as occlusions, noise, and background clutter in the point clouds, distinguishing ScanObjectNN from other simulated datasets and making it more difficult for classification tasks.

For our experiments, we use the most challenging version of ScanObjectNN, denoted as PB_T50_RS. This version incorporates perturbations to increase the dataset's difficulty. The suffix "T50" represents translation, with the bounding box randomly shifted up to 50% of its size from the box centroid along each world axis. Additionally, the suffixes "R" and "S" denote rotation and scaling.

Evaluation Metrics. We consider two key performance metrics: Overall Accuracy (OA) and Mean Accuracy (mAcc). "OA" represents the percentage of correctly classified objects across all classes, providing an overall assessment of model performance. "mAcc" calculates the average accuracy across all classes, offering insights into the model's performance across individual categories.

4.2 Results

To demonstrate the effectiveness of our method, we integrated DA^2 with various famou 3D classification models, including PointNet [19], PointNet++ [21], DGCNN [28], PointMLP [17], and PointNeXt [22]. We conducted experiments across three augmentation strategies: "Plain", "CDA", and "Ours". "Plain" represents training on the original dataset without any augmentation. "CDA" refers to conventional data augmentation, including global scaling, rotation, and translation with point-wise jittering [19]. "Ours" uses DA^2. These experiments were conducted using the PB_T50_RS dataset for training and validation.

Table 2. Classification results on ScanObjectNN with different classifiers using various augmentation strategies. The best accuracy results are highlighted in bold for easy identification.

Classifier	Plain		CDA		Ours	
	OA	mAcc	OA	mAcc	OA	mAcc
PointNet (2019)	66.48	60.49	73.56	69.30	**75.29** (1.73 ↑)	**71.67** (2.37 ↑)
PointNet++ (2019)	82.51	80.29	86.02	84.39	**86.61** (0.59 ↑)	**84.48** (0.09 ↑)
DGCNN (2019)	82.37	80.59	85.60	83.88	**86.29** (0.69 ↑)	**84.64** (0.76 ↑)
PointMLP (2022)	85.81	83.85	86.40	84.39	**86.47** (0.07 ↑)	**84.70** (0.31 ↑)
PointNeXt (2022)	83.66	80.74	87.06	85.24	**88.55** (1.49 ↑)	**87.37** (2.13 ↑)

Table 2 shows that all baseline 3D classifiers exhibited performance improvements when trained with our DA^2. Notably, PointNeXt achieved OA and mAcc of 88.55% and 87.37%, representing improvements of 1.49% and 2.13% compared to conventional data augmentation.

Table 3 compares several state-of-the-art point cloud data augmentation methods. We selected representative augmentation methods from various categories, including PointWOLF [12] for deformation, RSMix [13] for mixup, and AdaptPoints [27] for auto augmentation. Using PointNeXt as the baseline model, we applied these augmentation methods to the PB_T50_RS dataset for training. The experimental results indicate that our method outperforms other methods.

Table 4 presents the effectiveness of our augmentation on different datasets. We tested on two datasets: ScanObjectNN, which consists of scanned real-world indoor objects, and ModelNet40 [29], a point cloud object dataset derived from CAD models. Here, we used PointNet++ and PointNeXt as the classification backbones. The results on ScanObjectNN were consistent with those in Table 2.

Table 3. Comparisons were conducted with existing methods on the ScanObjectNN PB_T50_RS dataset, employing PointNeXt as the baseline classifier. In this comparison, 'S', 'R', 'T', 'D', 'J', 'Deform', and 'Mix' denote Scaling, Rotation, Translation, Point-wise Dropout, Jittering, Deformation, and Mixup, respectively. Augmentations enclosed in brackets signify that the operations are controlled by an auto-augmenter.

Method	**Augmentation**	**OA**	**mAcc**
PointNeXt		83.66	80.74
PointNeXt + CDA	S+R+T+J	87.06	85.24
PointNeXt + PointWOLF [12]	S+R+T+J+Deform	87.89	85.50
PointNeXt + RSMix [13]	S+R+T+J+Mix	88.27	87.02
PointNeXt + AdaptPoint [27]	S+R+T+J+(Deform+D)	88.10	86.60
PointNeXt + Ours	**S+R+T+(J+Deform+D)**	**88.55**	**87.37**

Table 4. Experiments on ScanObjectNN and ModelNet40 datasets, using CDA or DA^2 for augmentation. PointNet++ and PointNeXt were used as baseline classifiers.

Dataset	**Classifier**	**CDA**		**Ours**	
		OA	mAcc	OA	mAcc
ScanObjectNN [26]	PointNet++	86.02	84.39	**86.61**	**84.48**
	PointNeXt	87.06	85.24	**88.55**	**87.37**
ModelNet40 [29]	PointNet++	91.67	89.90	**92.42**	**90.32**
	PointNeXt	92.54	90.02	**92.71**	**90.60**

On ModelNet40, PointNet++ improved by 0.75% in OA and 0.42% in mAcc, while PointNeXt achieved gains of 0.17% in OA and 0.58% in mAcc. These results suggest that our method significantly aids in learning features from real-world objects and enhances the classifier's average performance across all categories.

4.3 Visualize Augmented Samples in Feature Space

As shown in Fig. 4, we analyzed and visualized the augmented samples within the feature space to elucidate the behaviors of DA^2 and CDTS. Furthermore, this visualization shows the comparison of our method and the CDA method, which showcases the utility of DA^2 in data exploration. Specifically, by projecting the augmented point clouds into the feature space, we tracked the augmentation trajectory of sample features across different categories. For illustration, we visualized one representative sample from each of the five categories. Each sample's augmented version was displayed for 10 consecutive epochs during training within the feature space. We can see the augmented samples become more and more challenging. Additionally, we noted that CDTS offers a mechanism to reset over-augmented samples, preventing the augmenter from generating out-of-distribution samples and enabling the augmenter to explore other augmentation

pathways. As a result, our proposed method gradually augments diverse samples, whereas CDA struggles to fully explore the augmentation space.

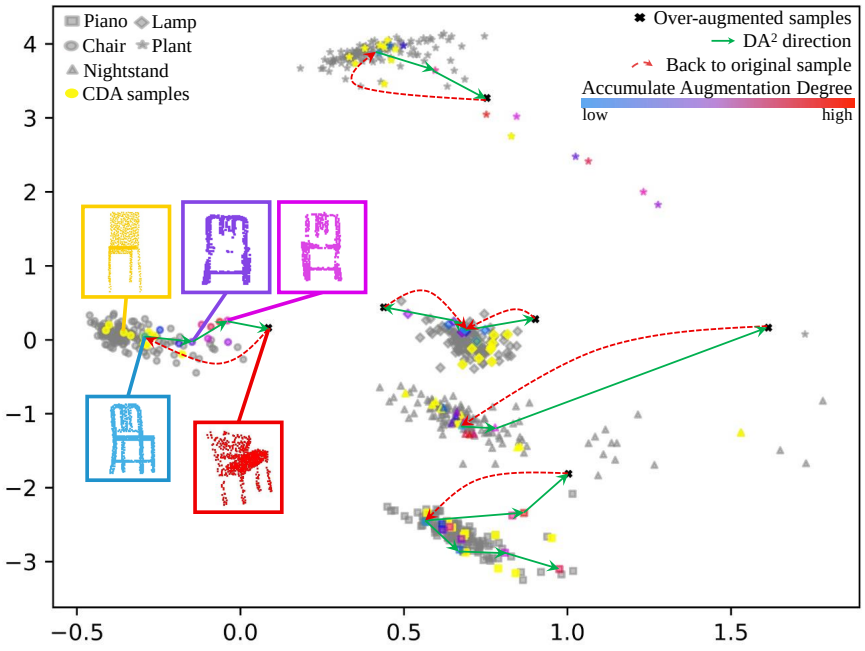

Fig. 4. We visualize the feature space extracted by DA^2 and CDA methods for comparison. Gray points represent samples from the original dataset, while the colored points denote the augmented samples.

4.4 Ablation Studies

Contributions of Components. We conducted ablation studies to assess the individual contributions of each component within our method, as detailed in Table 5. The first row of the table shows the scenario where only the Teacher-guided Auto Augmenter (TA), coupled with the teacher-student adversarial loss $\mathcal{L}_{T\_S}$, is utilized to generate augmented training samples. In this configuration, augmentations are not accumulated. Next, when DA^2 alone is applied, the absence of restrictions on accumulative augmentation leads to increasingly challenging samples, which hinders correct learning by the model. The integration of CDTS introduces a filtering and stopping mechanism that significantly enhances performance. The losses $\mathcal{L}_{SWD}$ and $\mathcal{L}_g$ synergistically improve outcomes, with the optimal performance observed when both are employed concurrently. Employing just one of the two losses, however, results in a more constrained impact on the model's performance.

Analysis on Different Thresholds. Table 6 presents experiments on the impact of different thresholds. The results of using fixed thresholds, 0, 0.5, and 0.7, show that without dynamically adjusting the threshold for different classes, it is difficult to balance improvements across all classes, resulting in minimal performance gains. Specifically, all samples are filtered out when the threshold is set to 0.9. When using the prediction score to determine the dynamic threshold (i.e., Dynamic mode), even without threshold ratio adjustments (i.e., curriculum), the performance is better. Furthermore, incorporating curriculum learning allows the model to learn progressively, leading to significant improvements.

TA	$\mathcal{L}_{T\_S}$	DA^2	CDTS	$\mathcal{L}_{SWD}$	$\mathcal{L}_g$	OA	mAcc
✓	✓					87.37	85.81
✓	✓	✓				85.67	82.65
✓	✓	✓	✓			86.78	84.45
✓	✓	✓	✓	✓		86.81	84.80
✓	✓	✓	✓		✓	87.68	85.46
✓	✓	✓	✓	✓	✓	**88.55**	**87.37**

Table 5. Contributions of Components.

Threshold	Dynamic	Curriculum	OA	mAcc
0			87.30	85.60
0.5			87.37	85.48
0.7			87.23	84.98
0.9			87.16	85.01
-	✓		87.86	85.74
-	✓	✓	**88.55**	**87.37**

Table 6. Analysis on different threshold.

5 Conclusion

This paper introduces Degree-Accumulated Data Augmentation (DA^2), a novel data augmentation strategy for 3D point clouds. By accumulating augmentations, we overcome the limitations of traditional methods that constrain the exploration space. To the best of our knowledge, this is the first study to employ this technique. Our approach combines a teacher-guided auto augmenter to generate reasonable augmented samples at each step and a Curriculum Dynamic Threshold Selection (CDTS) to further filter suitable training samples from the student's perspective. This design ensures that, despite the increased difficulty and variability of samples due to accumulation, the smooth learning curve allows the model to learn high-level features. Finally, we demonstrate through extensive experiments and analyses that our method effectively enhances model performance.

Acknowledgements. This work was financially supported in part (project number: 112UA10019) by the Co-creation Platform of the Industry Academia Innovation School, NYCU, under the framework of the National Key Fields Industry-University Cooperation and Skilled Personnel Training Act, from the Ministry of Education (MOE) and industry partners in Taiwan. It is also supported in part by the National Science and Technology Council, Taiwan, under Grant NSTC-112-2221-E-A49-089-MY3, Grant NSTC-110-2221-E-A49-066-MY3, Grant NSTC-111- 2634-F-A49-010, Grant NSTC-112-2425-H-A49-001, and in part by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and the Ministry of Education (MOE), Taiwan. We sincerely thank Hon Hai Research Institute for their invaluable support.

References

1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning. pp. 41–48 (2009)
2. Bonneel, N., Rabin, J., Peyré, G., Pfister, H.: Sliced and radon wasserstein barycenters of measures. Journal of Mathematical Imaging and Vision **51**, 22–45 (2015)
3. Chen, Y., Hu, V.T., Gavves, E., Mensink, T., Mettes, P., Yang, P., Snoek, C.G.: Pointmixup: Augmentation for point clouds. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 330–345. Springer (2020)
4. Cheng, S., Leng, Z., Cubuk, E.D., Zoph, B., Bai, C., Ngiam, J., Song, Y., Caine, B., Vasudevan, V., Li, C., et al.: Improving 3d object detection through progressive population based augmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16. pp. 279–294. Springer (2020)
5. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3075–3084 (2019)
6. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 113–123 (2019)
7. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 702–703 (2020)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Ho, D., Liang, E., Chen, X., Stoica, I., Abbeel, P.: Population based augmentation: Efficient learning of augmentation policy schedules. In: International conference on machine learning. pp. 2731–2741. PMLR (2019)
10. Hua, B.S., Tran, M.K., Yeung, S.K.: Pointwise convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 984–993 (2018)
11. Jaritz, M., Gu, J., Su, H.: Multi-view pointnet for 3d scene understanding. In: Proceedings of the IEEE/CVF international conference on computer vision workshops. pp. 0–0 (2019)
12. Kim, S., Lee, S., Hwang, D., Lee, J., Hwang, S.J., Kim, H.J.: Point cloud augmentation with weighted local transformations. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 548–557 (2021)
13. Lee, D., Lee, J., Lee, J., Lee, H., Lee, M., Woo, S., Lee, S.: Regularization strategy for point cloud via rigidly mixed sample. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15900–15909 (2021)
14. Li, R., Li, X., Heng, P.A., Fu, C.W.: Pointaugment: an auto-augmentation framework for point cloud classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6378–6387 (2020)
15. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. Advances in Neural Information Processing Systems **32** (2019)

16. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. Advances in neural information processing systems **32** (2019)
17. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. arXiv preprint arXiv:2202.07123 (2022)
18. Nguyen, T., Pham, Q.H., Le, T., Pham, T., Ho, N., Hua, B.S.: Point-set distances for learning representations of 3d point clouds. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10478–10487 (2021)
19. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
20. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5648–5656 (2016)
21. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
22. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: Pointnext: Revisiting pointnet++ with improved training and scaling strategies. Advances in Neural Information Processing Systems **35**, 23192–23204 (2022)
23. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 945–953 (2015)
24. Suzuki, T.: Teachaugment: Data augmentation optimization using teacher knowledge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10904–10914 (2022)
25. Umam, A., Yang, C.K., Chuang, Y.Y., Chuang, J.H., Lin, Y.Y.: Point mixswap: Attentional point cloud mixing via swapping matched structural divisions. In: European Conference on Computer Vision. pp. 596–611. Springer (2022)
26. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1588–1597 (2019)
27. Wang, J., Ding, L., Xu, T., Dong, S., Xu, X., Bai, L., Li, J.: Sample-adaptive augmentation for point cloud recognition against real-world corruptions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14330–14339 (2023)
28. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog) **38**(5), 1–12 (2019)
29. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
30. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16259–16268 (2021)
31. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4490–4499 (2018)