Attention Discriminant Sampling for Point Clouds

Cheng-Yao Hong[1] Yu-Ying Chou[2] Tyng-Luh Liu[1]
[1]Institute of Information Science, Academia Sinica, Taiwan [2]National Taiwan University
{sensible,liutyng}@iis.sinica.edu.tw d07922014@csie.ntu.edu.tw

Abstract

This paper describes an attention-driven approach to 3-D point cloud sampling. We establish our method based on a structure-aware attention discriminant analysis that explores geometric and semantic relations embodied among points and their clusters. The proposed attention discriminant sampling *(ADS) starts by efficiently decomposing a given point cloud into clusters to implicitly encode its structural and geometric relatedness among points. By treating each cluster as a structural component, ADS then draws on evaluating two levels of self-attention: within-cluster and between-cluster. The former reflects the semantic complexity entailed by the learned features of points within each cluster, while the latter reveals the semantic similarity between clusters. Driven by structurally preserving the point distribution, these two aspects of self-attention help avoid sampling redundancy and decide the number of sampled points in each cluster. Extensive experiments demonstrate that ADS significantly improves classification performance to* ***95.1%*** *on ModelNet40 and* ***87.5%*** *on ScanObjectNN and achieves* ***86.9%*** *mIoU on ShapeNet Part Segmentation. For scene segmentation, ADS yields* ***91.1%*** *accuracy on S3DIS with higher mIoU to the state-of-the-art and* ***75.6%*** *mIoU on ScanNetV2. Furthermore, ADS surpasses the state-of-the-art with* ***55.0%*** *mAP$_{50}$ on ScanNetV2 object detection.*

1. Introduction

Point cloud sampling is an essential step for designing practical solutions to the respective 3-D computer vision applications. While existing such techniques mostly rely on utilizing *distance-based* geometric information to yield efficient sampling, their main drawback is the lack of consideration of both the semantic and the structural aspects of the underlying point cloud distribution, as shown in Figure 1. We aim to introduce a *cluster-based* formulation that leverages the proposed discriminant attention analysis to implicitly achieve *structure-aware* point cloud sampling.

With the great success of deep learning in solving 2-D computer vision problems, many studies now have turned to dealing with realistic and challenging 3-D tasks. It is known that 3-D data can often be represented in various formats, including depth images, volume grids (voxels), multiple flat polygons (meshes), and point clouds. Among them, point clouds are raw data obtained by scanning objects with instruments, *e.g.*, LiDAR scanners and multibeam sonars. Depending on the functionality of the instrument, the point coordinates can be accompanied with other useful information such as colors and normal vectors. As the point cloud representation retains original geometry information without any distortion, it is suitable for various applications that require 3-D scene understanding, including autonomous navigation, virtual reality, robotics, etc. However, the irregular and unstructured data format often makes learning directly from raw 3-D point clouds inefficient. Previous approaches have addressed this issue of vast computation via preprocessing, for example, by projecting point clouds to depth images [37, 38] and enabling CNNs to tackle the resulting tasks. The other feasible reduction is to consider voxelization by simplifying the data into rasterized grid [27, 8, 49], which is applicable to 3-D convolutional networks. Although such attempts could ease the computation demand, they may still lead to unstable performance when dealing with intricate

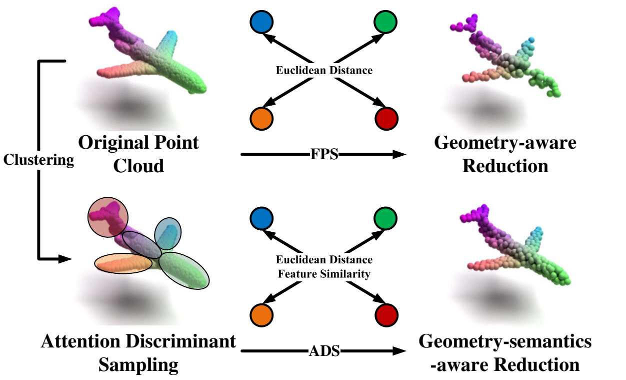

Figure 1. **Overview**. Unlike distance-based sampling strategies, *e.g.* Farthest Point Sampling (FPS), the proposed structure-aware Attention Discriminant Sampling (ADS) explores both geometric and semantic information to improve representation learning and sampling efficiency for 3-D point clouds.

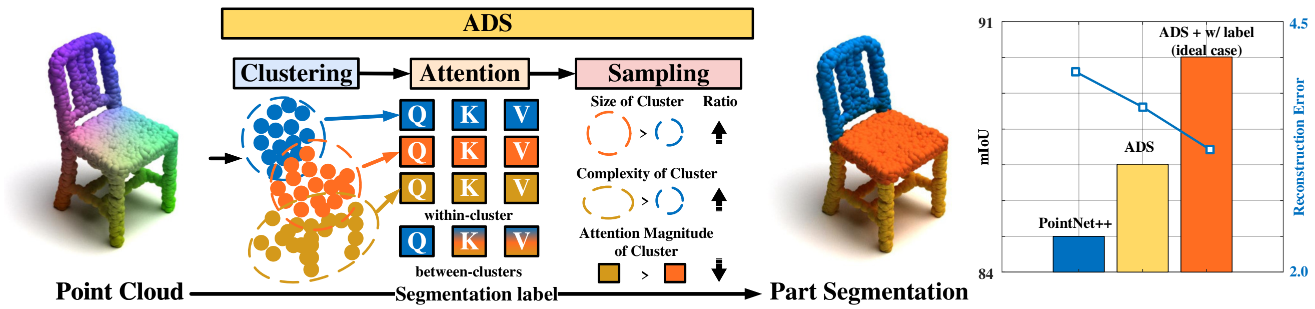

Figure 2. **The proposed Attention Discriminant Sampling (ADS) framework** comprises three main processes: clustering, attention, and sampling. Unlike the conventional Set Abstraction (SA) module, ADS initially groups the points via clustering, which serves to implicitly identify parts or objects within the entire scene. The subsequent sampling step is then cluster-wise performed by taking account of the attention responses from within-cluster and between-cluster evaluations, the size, and the complexity of each cluster to determine the respective sampling ratio. Taking, for example, sampling with ADS as input for part segmentation, we show the performance gain (yellow bar) and a performance upper bound (orange bar) when replacing the clustering outcome in ADS with exact information of part labels.

scenes or objects of complex shapes.

As one of the pioneer frameworks on learning with point clouds, PointNet [30] is developed to extract point features directly from the raw data. It overcomes the unordered property of a point cloud via the use of a simple symmetric function (*i.e*., max pooling), and yields invariant representation through the proposed T-net with coordinates transformation. However, PointNet does not model local context well and may not capture objects of various scales properly. The improved version, PointNet++ [32], is therefore introduced; it employs hierarchical learning architecture that consists of several *set abstraction* layers. Such layers apply sampling and grouping to abstract local regions and achieve the balance between efficiency and performance. From the two frameworks and their related applications, the effective use of point cloud sampling to the end results of downstream tasks has emerged as a crucial factor. While the majority of point cloud techniques, including [30, 32], adopt *farthest point sampling* (FPS) [29], we are motivated to develop a point-cloud sampling strategy beyond the perspective of distance-based reasoning.

The main idea behind our approach is motivated by the observation that point cloud sampling could better represent the original raw data if information about the structural components and complexities of the underlying object has been made available. Suppose that for each object part we know its corresponding points, the semantic complexity, and relatedness to other components. Such information can then be used to decide how to apportion the specified number of points for sampling, while retaining its object structure properly. In practice, the structural information is often not present; however, it is reasonable to group the points into clusters and design an efficient cluster-based sampling scheme that is designed to conceptually approximate the above-mentioned ideal case. To simultaneously explore semantic and geometric properties of a point cloud, we introduce *attention discriminant sampling* (ADS) that leverages feature similarities with spatial relationships to select those points that are representative in both aspects and induce less redundancy. As shown in Figure 1, the proposed ADS first divides a point cloud into clusters, and enables the algorithm to consider sampling at point-wise and cluster-wise levels. Our method then calculates *self-attention* relations of the points within each cluster and between clusters to determine how the sampled points are distributed among the clusters. Unlike other attention-based 3-D approaches [52, 59] that only consider the points within each subset obtained by *farthest point sampling* (FPS) [29], ADS assesses semantic and geometric relations of the points *locally* (within each cluster) and *globally* (between clusters) and utilizes these self-attention responses to achieve effective and representative point-cloud sampling. Figure 2 illustrates the advantage of the ADS scheme for the downstream task of part segmentation, compared with the performance upper bound obtained when part information is provided for the sampling step.

2. Related work

3-D point cloud representation. High-quality representation is crucial to achieve good performance for 3-D point cloud tasks. According to the central data format, relevant literature can be divided into image-based, voxel-based, and point-based. Image-based approaches, such as MVCNN [39], SimpleView [11], and MVTN [12], transfer a 3-D point cloud into various 2-D views to learn their underlying model. On the other hand, 3DShapeNet [49], VoxNet [27], Subvolume [31] and O-CNN [43] map points into voxels for solving the tasks with 3-D convolutional networks. To alleviate representation constraints, point-based methods, *e.g*., PointNet++ [32], RepSurf [35], PointMLP [26], PointNeXt [33] and MaskPoint [21], deal with original unordered points. Our work also learns the representation directly from point clouds in that the raw data include all the available information and can be coupled with a task-specific head for end-to-end training.

Point cloud sampling. Due to the computational demand, subset sampling for point cloud applications is a key step for practical implementations. The FPS [29], which is agnostic to downstream tasks, is considered the de facto technique, and has been widely used in mainstream frameworks, *e.g*., PointNet++ [32]. Chen *et al*. [5] further consider semantic features to improve FPS. To couple the sampling procedure with specific downstream tasks, learnable sampling models have been established to account for the task-specific objectives. Take, for example, that Dovrat *et al*. propose a data-driven sampling model that yields sampled subsets by optimizing with respect to the reconstruction error and the downstream task loss [9]. In [19], Lang *et al*. introduce the SampleNet model that explores differentiable relaxation to produce sampling points from the weighted average of input points. Cheng *et al*. [6] develop a meta-sampler that can rapidly adapt to various datasets or tasks with a meta-learning technique. Still, the lack of sampling techniques to comprehensively take account of both geometric and semantic relations of a point cloud could be an issue for further advancing the progress of this emerging research field.

Self-attention on point clouds. Transformer-based techniques have been recently applied in solving various point cloud applications, including object detection [13, 3, 57, 45], segmentation [18, 46] and registration [53, 34]. The Point-Transformer series [59, 48] utilizes a subtraction-based attention block to improve classification and segmentation. PAT [52] proposes *group shuffle attention* to replace multi-head attention for learning point-wise relationship efficiently. To expand the receptive field of the transformer, He *et al*. [13] develop a voxel-based set transformer that achieves linear complexity to the number of points. In [18], the authors introduce a hierarchical structure to enlarge the receptive field and shift windows to attain interactive information between windows. Our method instead considers establishing within-cluster and between-cluster self-attention blocks for retaining local and global features of a point cloud, and achieves structure-aware representative sampling from the geometric and semantic perspectives.

3. Our method

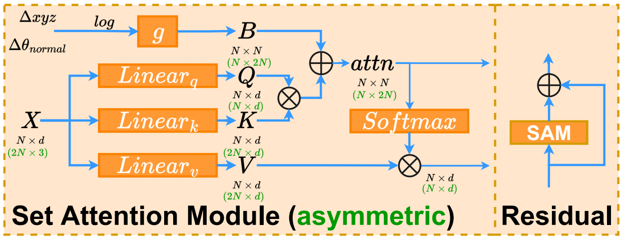

Figure 3. The architecture of Set Attention Module (SAM). The asymmetric version of SAM is also included and marked in green.

Different from the widely adopted FPS, the proposed sampling method considers not only the Euclidean distance based geometric relation between 3-D points but also the self-attention induced semantic relation from their features. More importantly, to implicitly retain the structural information, we consider a *cluster-based* formulation where in the ideal scenario the resulting point clusters would correspond to the parts of an underlying object. To this end, we first group a given point cloud into clusters and then compute *within-cluster* and *between-cluster* self-attention to determine the number of points to be sampled and the selection outcome in each cluster. This way we establish *attention discriminant sampling* (ADS) and achieve a compact and representative reduction of a dense point cloud for downstream tasks.

3.1. Problem setting

We represent a given point cloud as a set of 3-D points, $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{n} \in \mathbb{R}^{n\times 3}$, and n is the total number of points. As n is typically large for most applications of interest, our goal is to establish a general and effective sampling scheme to extract from $\mathcal{P}$ an m-point *representative* subset,

$$\hat{\mathcal{P}} = \{\mathbf{p}_{\hat{i}_1}, \dots, \mathbf{p}_{\hat{i}_m}\} \subseteq \mathcal{P}, \tag{1}$$

where $|\hat{\mathcal{P}}| = m \ll n$ and the subscript $\hat{i} \mapsto j \in \{1, \dots, n\}$ is specified by the resulting one-to-one index mapping. Among the numerous ways to carry out point cloud sampling, a sampling scheme and the resulting $\hat{\mathcal{P}}$ are preferable if they respect the following useful properties.

- The m sampled points in $\hat{\mathcal{P}}$ are obtained by preserving object structure and reducing redundancy in the context of both geometric and semantic perspectives.
- For each reasonable value of m, the more effective a sampling scheme is, the better reconstruction result can be achieved from $\hat{\mathcal{P}}$ to the original point cloud $\mathcal{P}$.
- The design of point cloud sampling is generic and can be readily plugged into various network models for a broad spectrum of downstream tasks.

3.2. Set attention module

As the self-attention mechanism is the key operation throughout our network model, we begin by describing our *set attention module* (SAM), and the details can be found in Figure 3. Analogous to [42], SAM first projects a set of input features $X = \{\mathbf{x}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times d}$ to three different embeddings, namely, $Q = \{\mathbf{q}_i\}_{i=1}^N$, $K = \{\mathbf{k}_i\}_{i=1}^N$, and $V = \{\mathbf{v}_i\}_{i=1}^N$ in $\mathbb{R}^{N\times d}$. (Note that $\mathbf{x}_i$ is the feature vector of point $\mathbf{p}_i$.) The self-attention over X and the resulting features are computed by

$$\mathrm{attn}(X) = \frac{QK^T}{\sqrt{d}} + B \in \mathbb{R}^{N\times N}, \quad (2)$$

$$\mathrm{SAM}(X) = \mathrm{softmax}(\mathrm{attn}(X))V \in \mathbb{R}^{N\times d}, \quad (3)$$

where $B = [b_{ij}]$ includes relative bias factors b_{ij} between $\mathbf{p}_i$ and $\mathbf{p}_j$. Inspired by [23], we add a two-layer MLP, denoted as $g(\cdot)$, to predict the bias factor instead of adopting a lookup table. To fully explore the relative geometric information between each pair of points, the input to the bias network module g includes their relative position and the included angle between the normal vectors. We have

$$B = g(\,\log(\Delta x, \Delta y, \Delta z, \Delta\theta)\,) \in \mathbb{R}^{N\times N}, \quad (4)$$

where $\Delta x, \Delta y, \Delta z$ and $\Delta\theta$ represent all the pairwise offsets of relative positions and relative normal angles, *i.e.*, $\arccos(\mathrm{Normal}(\mathbf{p}_i), \mathrm{Normal}(\mathbf{p}_j))$, respectively.

3.3. Point cloud clustering

The first step of our approach is to divide a point cloud into clusters. We consider *mean shift clustering* [7] to group the point set into distinct modes and take these cluster centers as the initial seeds for subsequent analysis. At each iteration of clustering, the algorithm calculates the so-called mean-shift vector $\mathbf{m}(\mathbf{p})$ for each point $\mathbf{p}$ by

$$\mathbf{m}(\mathbf{p}) = \frac{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} G(\mathbf{p}_i - \mathbf{p})\mathbf{p}_i}{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} G(\mathbf{p}_i - \mathbf{p})}, \quad (5)$$

where G is a Gaussian kernel with the bandwidth parameter h, and $\mathcal{N}(\mathbf{p})$ comprises a specified neighborhood of point $\mathbf{p}$. With the expression of mean-shift vector in (5), the iterative clustering updates each point $\mathbf{p}$ as follows:

$$\mathbf{p} \leftarrow \mathbf{p} + \mathbf{m}(\mathbf{p}). \quad (6)$$

The algorithm iterates the above two steps (5) and (6) until the convergence of a mean-shift vector. For a given point cloud $\mathcal{P} = \{\mathbf{p}_1, ..., \mathbf{p}_n\}$, the mean-shift clustering divides $\mathcal{P}$ into K clusters with the center set, $\mathcal{O} = \{\mathbf{p}_{o_1}, \ldots, \mathbf{p}_{o_K}\}$. In all the experiments, our implementation sets the bandwidth h to 1.5 cm and the maximum number of mean-shift iterations to 300. For simplification, we restrict the maximum number of clusters of a point set $\mathcal{P}$ to be less or equal to $m/4$.

In an ideal scenario that the point clusters by mean shift did correspond to object parts or instances of *thing* and *stuff* in a scene, the effectiveness of sampling could then be significantly boosted by taking account of such useful information, as depicted in Figure 2. The practical outcome is far from the ideal case in that these clusters are obtained solely based on the geometric relation of distances between points, but neglecting the semantic relation from the underlying features. Nevertheless, the procedure of point grouping enables our cluster-based formulation and plays a central role in the proposed *attention discriminant sampling* (ADS).

3.4. Cluster discriminant attention

The resulting K clusters empower designing attention-driven sampling beyond the point level. Specifically, we connect the sampling strategy with two aspects of self-attention: *within-cluster* and *between-cluster*. The former reflects the semantic complexity entailed by the learned features of points within each cluster, while the latter reveals the semantic similarity between clusters. It turns out that the two kinds of attention responses are informative for deciding how our algorithm achieves point cloud sampling.

Within-cluster self-attention. Discriminant attention analysis within each cluster is to determine its semantic complexity from the distribution of self-attention responses. To efficiently carry out such evaluation, we consider, for each cluster center $\mathbf{p}_{o_c}$, a k-point neighborhood $\mathcal{N}_k(\mathbf{p}_{o_c})$ and compute its set attention matrix $\mathrm{attn}(\mathcal{N}_k(\mathbf{x}_{o_c})) \in \mathbb{R}^{k\times k}$ as in (2). By taking average pooling along the column direction, we obtain a within-cluster attention vector $A_W(\mathbf{p}_{o_c})$ of k entries, each of which reflects a point's average semantic relatedness to all points in $\mathcal{N}_k(\mathbf{p}_{o_c})$. To quantitatively measure the semantic complexity of cluster c, we compute from $A_W(\mathbf{p}_{o_c})$ the standard deviation σ_c for the distribution of self-attention responses. In the ADS scheme, σ_c is positively correlated to the number of points to be sampled from cluster c. Notice that although the above computation is performed cluster-wise, we deploy a weight-sharing SAM module over the K clusters to retain a compact network model.

Between-cluster self-attention. We first need to come up with a representative feature vector for each cluster c. Instead of simply using the feature vector $\mathbf{x}_{o_c}$ of the cluster center $\mathbf{p}_{o_c}$, we prefer a more robust feature representation that accounts for the underlying feature distribution. To this end, we exploit the available within-cluster evaluation and obtain k aggregated feature vectors from $\mathrm{SAM}(\mathrm{attn}(\mathcal{N}_k(\mathbf{x}_{o_c}))) \in \mathbb{R}^{k\times d}$. We then perform average pooling over the k features to yield the global feature vector $\mathbf{f}_c \in \mathbb{R}^d$ for cluster c.

To compute between-cluster self-attention, we again apply the SAM operation over the set of K global feature vectors and derive the set attention matrix $\mathrm{attn}(\{\mathbf{f}_c\}_{c=1}^K) \in \mathbb{R}^{K\times K}$. We assess the semantic similarity of cluster c to other clusters by summing the self-attention values in row c of the between-class attention matrix. This step results in a set of semantic similarity measures $\{s_c\}_{c=1}^K$. Opposite to the effect of σ_c, the value of s_c is negatively correlated to the number of points to be sampled from cluster c since it reflects the degree of semantic redundancies among clusters.

The robustness of learning within-cluster and between-cluster self-attention can be consolidated by adding a regularization effect on the K cluster-wise (global) feature vectors $\{\mathbf{f}_c\}_{c=1}^{K}$ as they form the bridge between the two aspects of computation. Our motivation is that if these feature vectors are sufficiently representative, it should be possible to obtain from them a reasonable reconstruction $\tilde{\mathcal{P}}$ to the original point cloud $\mathcal{P}$. To realize this intuition, we average the K feature vectors to yield $\bar{\mathbf{f}}$ and then generate $\tilde{\mathcal{P}}$ by projecting and reshaping. These steps can be expressed by

$$\{\mathbf{f}_c\} \xrightarrow{\psi} \{\psi(\mathbf{f}_c)\} \xrightarrow{\text{mean}} \bar{\mathbf{f}} \in \mathbb{R}^{3n} \xrightarrow{\text{reshape}} \tilde{\mathcal{P}} \in \mathbb{R}^{n\times 3} \tag{7}$$

where $\psi : \mathbb{R}^{K\times d} \to \mathbb{R}^{K\times 3n}$ is 1-D depthwise convolution. With (7), we apply Earth Mover's Distance (EMD) as in [10] to establish the following reconstruction loss:

$$\mathcal{L}_{\text{EMD}} = \text{EMD}(\tilde{\mathcal{P}}, \mathcal{P}). \tag{8}$$

3.5. Attention discriminant sampling

We are now ready to describe how ADS performs effective sampling over the given point cloud $\mathcal{P}$. Recall that $\mathcal{P}$ is of size n and the sampling process yields $\hat{\mathcal{P}}$ of size m. Based on our analyses of cluster discriminant attention, we determine the number of points to be sampled from each cluster c via calculating the following sampling ratio,

$$w_c = \frac{1}{\Sigma}\,(n_c \times \sigma_c \times 1/s_c) \tag{9}$$

where n_c is the size of cluster c and Σ is the normalization factor to ensure $\sum w_c = 1$. The number of points to be sampled from cluster c can be denoted as $m_c = m \times w_c$.

ADS cluster-wise sampling. We carry out cluster-wise sampling according to a decreasing order of m_c. Consider now for cluster c, the m_c points to be sampled will be chosen from $\mathcal{N}_k(\mathbf{p}_{o_c})$. Our formulation implicitly assumes $k \geq m_c$ and purposely sets the value of k to ensure the assumption is valid. Even when m_c is larger than k, we could simply sample k points from the underlying cluster and add $m_c - k$ to the $m_{c'}$ of the succeeding cluster c'. (Such an exception has not been encountered in all our experiments.)

To determine which m_c points are to be sampled from cluster c, we additionally extend the SAM to an asymmetric form. (See Figure 4.) In particular, besides $\mathcal{N}_k(\mathbf{p}_{o_c})$ of cluster c, we randomly select the other set of k points, denoted as X'_c, from other clusters. While all the $2k$ points are used to generate keys and values, only the k points in $\mathcal{N}_k(\mathbf{p}_{o_c})$ are taken as queries. We express the resulting asymmetric attention matrix by

$$\hat{A}_c = \widehat{\text{attn}}(\mathcal{N}_k(\mathbf{p}_{o_c}) \cup X'_c) \in \mathbb{R}^{k\times 2k} \tag{10}$$

where $\widehat{\text{attn}}$ symbolizes the asymmetric set attention (weight sharing across K clusters). Each row in $\hat{A}_c$ corresponds to the outcomes of using a point in $\mathcal{N}_k(\mathbf{p}_{o_c})$ as the query to all $2k$ keys. Expanding the keys to exceeding cluster c is advantageous as it *globally* evaluates each query and produces more robust attention evidence. It is worth mentioning that part of the calculation in (10) has already been completed in evaluating the within-class attention when keys are generated by the points within cluster c . We obtain aggregated attention of each row by summing all these $2k$ attention values. The steps to sample the m_c points are listed below.

1. The first point to be sampled is the one with the largest value of aggregated attention from $\hat{A}_c$.
2. Let S include the indices for those points that have already been sampled, the next point to be sampled is decided by the following returned index:

$$i^* \leftarrow \min_{i \notin S} \sum\nolimits_{j\in S} \hat{A}_c(i.j). \tag{11}$$

The step 2 by (11) is repeated until ADS has sampled the required m_c points for cluster c. We now justify the reasoning of the proposed sampling method. In estimating the number of points to be sampled, ADS considers the cluster size n_c, the within-cluster semantic complexity σ_c, and the between-cluster semantic similarity s_c as in (9). In addition, the sampling within each cluster begins with the most attending point (of the largest aggregated attention value) and then proceeds in a *least attended point* fashion (as in (11)). Finally, we include a second regularization loss $\mathcal{L}_{\text{Sample}}$ to facilitate feature consistency between the cluster feature vector $\bar{\mathbf{f}}$ and the average feature vector from the sampled points of each cluster. We have

$$\mathcal{L}_{\text{Sample}} = \sum\nolimits_{c=1}^{K} \|\mathbf{f}_c - \text{mean}(\hat{\mathbf{f}}_{S_c})\|_2 \tag{12}$$

where S_c and $\hat{\mathbf{f}}_{S_c}$ respectively represent the set of sampled points from cluster c and their feature vectors generated by the asymmetric set attention described in (10).

Total loss. We express the total loss of learning ADS as

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{EMD}} + \mathcal{L}_{\text{Sample}} + \mathcal{L}_{\text{Target}} \tag{13}$$

where $\mathcal{L}_{\text{Target}}$ is the loss to be specified by the targeted downstream task.

3.6. Model architecture

As shown in Figure 4, the network architecture of ADS is similar to PointNet++ [32], which adopts staked modules. In this work, we provide three practical designs of ADS for solving three types of downstream tasks, including classification, part/semantic segmentation, and object detection.

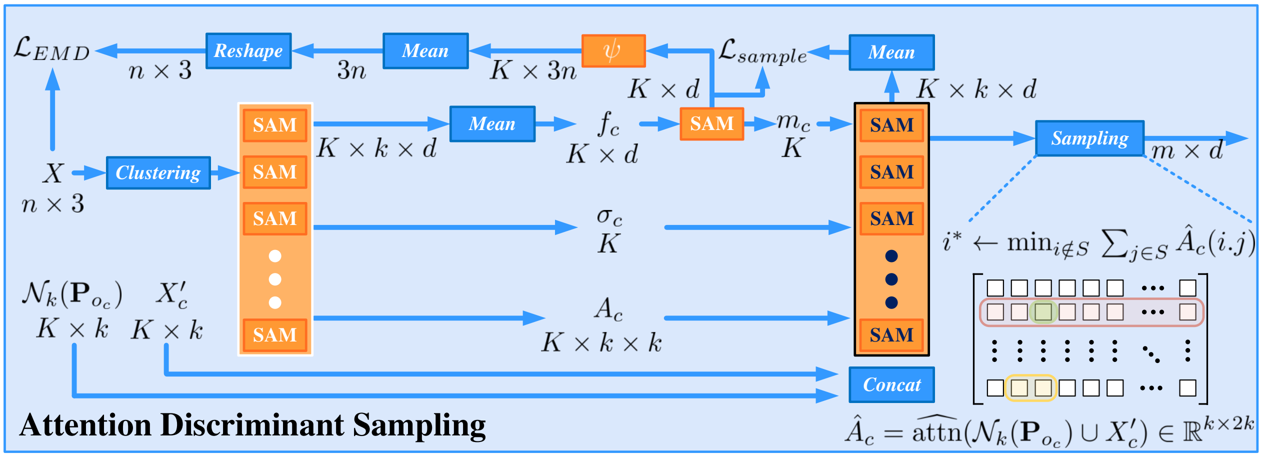

Figure 4. **The ADS architecture.** It begins by dividing the input point cloud into clusters. Then, the ADS explores within-cluster and between-cluster self-attention to decide the sampling outcome. Note that the asymmertric version of set attention module is marked in black.

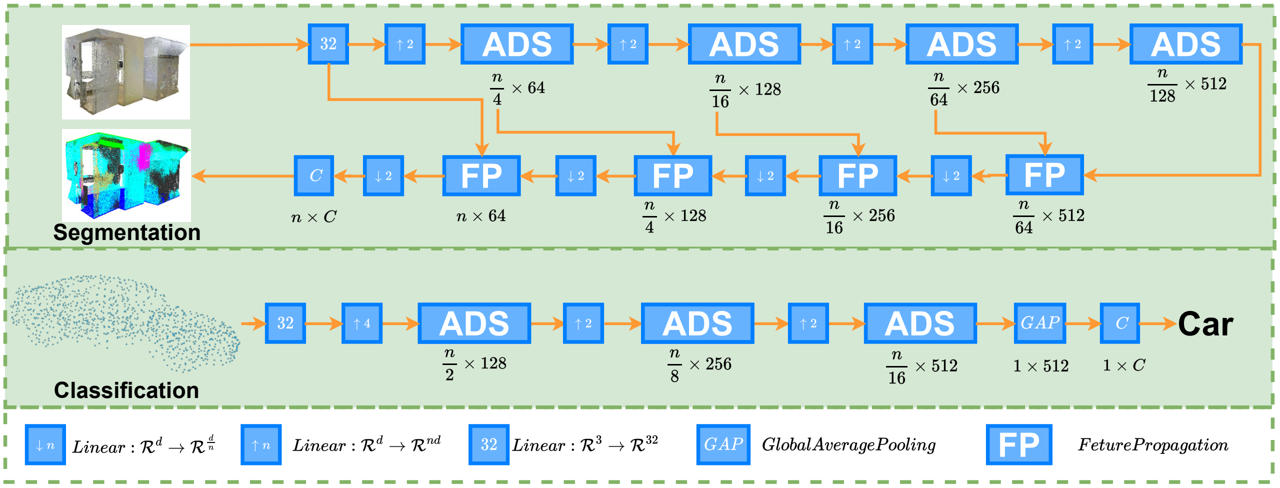
32
↑2
ADS
$\frac{n}{4} \times 64$
↑2
ADS
$\frac{n}{16} \times 128$
↑2
ADS
$\frac{n}{64} \times 256$
↑2
ADS
$\frac{n}{128} \times 512$
Segmentation
C
↓2
FP
↓2
FP
↓2
FP
↓2
FP
$n \times C$
$n \times 64$
$\frac{n}{4} \times 128$
$\frac{n}{16} \times 256$
$\frac{n}{64} \times 512$
Classification
32
↑4
ADS
$\frac{n}{2} \times 128$
↑2
ADS
$\frac{n}{8} \times 256$
↑2
ADS
$\frac{n}{16} \times 512$
GAP
1×512
C
$1 \times C$
Car
↓n $Linear : \mathcal{R}^d \rightarrow \mathcal{R}^{\frac{d}{n}}$
↑n $Linear : \mathcal{R}^d \rightarrow \mathcal{R}^{nd}$
32 $Linear : \mathcal{R}^3 \rightarrow \mathcal{R}^{32}$
GAP GlobalAveragePooling
FP FeturePropagation

Figure 5. **The two network architectures with ADS for point-cloud classification and segmentation**. The classification model consists of three ADS modules, and the segmentation architecture includes four ADS modules and feature propagation modules.

We illustrate the two respective network architectures for the downstream tasks of classification and segmentation in Figure 5, while the network model for the detection task is only slightly different from the one for segmentation.

Classification model. The point cloud classification network comprises three ADS modules, with n/2, n/8, and n/16 sampled points, and feature dimensions of 128, 256, and 512, respectively. The numbers of selected points k in ADS are 16, 32, and 64. After global average pooling, the features pass through a linear layer that projects the dimension from 512 to C, the number of classes, to predict the label.

Segmentation model. The part/semantic segmentation network includes four ADS modules with n/4, n/16, n/64, and n/128 sampled points, and feature dimensions of 64, 128, 256, and 512, respectively. The numbers of selected points k in ADS are 8, 16, 32, and 64. The features then pass through a decoder with four feature propagation modules, as in PointNet++, to achieve the up-sampling operation.

Detection model. For the detection task, we adopt Group-Free [25] as the framework and replace the backbone with ADS to obtain the detection outcomes on ScanNetV2. The proposed object detection architecture comprises four ADS modules with 2048, 1024, 512, and 256 sampled points from 50,000 points. The numbers of selected points k in ADS are 8, 16, 32, and 64, respectively.

4. Experimental results

We justify the usefulness of ADS with the classification, segmentation and detection tasks. For classification, we

Method	**# points**	**input**	**ModelNet40**	
			OA (%)	**mAcc (%)**
MVCNN [39]	-	Image	90.1	-
SimpleView [11]	-	Image	93.0	90.5
MVTN [12]	-	Image	93.5	92.2
3DShapeNet [49]	-	Voxel	77.3	84.7
VoxNet [27]	-	Voxel	85.9	83.0
Subvolume [31]	-	Voxel	89.2	87.2
O-CNN [43]	-	Voxel	90.6	-
PointNet [30]	1024	Points	89.2	86.0
PCNN [2]	1024	Points	92.3	-
PointCNN [20]	1024	Points	92.5	88.8
DGCNN [44]	1024	Points	92.9	90.2
RS-CNN [22]	1024	Points	93.6	-
PointNeXt-S [33]	1024	Points	93.2	90.8
PointMLP [26]	1024	Points	94.1	91.3
ADS	1024	Points	**94.3**	**91.7**
PAT [52]	1024	Points w/ Normals	91.7	-
PointNet++ [32]	1024	Points w/ Normals	91.9	90.3
PointConv [47]	1024	Points w/ Normals	92.5	-
PointASNL [51]	1024	Points w/ Normals	93.2	-
PointTransformer [59]	1024	Points w/ Normals	93.7	90.6
RepSurf-T [35]†	1024	Points w/ Normals	94.0	91.1
RPNet-W9 [36]	1024	Points w/ Normals	94.1	-
PointTransformer-V2 [48]	1024	Points w/ Normals	94.2	91.6
RepSurf-U [35]†	1024	Points w/ Normals	94.4	91.4
ADS	1024	Points w/ Normals	**95.1**	**92.3**

Table 1. **Classification on ModelNet40**: ADS versus others. OA denotes overall accuracy, mAcc represents the mean of per-class accuracy, and † indicates without using multiscaling inference.

Method	**# points**	**input**	**ScanObjectNN**	
			OA (%)	**mAcc (%)**
SimpleView [11]	-	Image	79.5	-
MVTN [12]	-	Image	82.8	-
PointNet [30]	1024	Points	68.2	63.4
DGCNN [44]	1024	Points	78.1	73.6
PointCNN [20]	1024	Points	78.5	-
PointMLP [26]	1024	Points	85.4	83.9
PointNeXt-S [33]	1024	Points	**87.7**	**85.8**
PointNet++ [32]	1024	Points w/ Normals	77.9	75.4
RepSurf-T [35]	1024	Points w/ Normals	84.1	81.2
RepSurf-U [35]	1024	Points w/ Normals	84.3	81.3
ADS	1024	Points w/ Normals	87.5	85.1

Table 2. **Classification on ScanObjectNN**: ADS versus others.

Method	**S3DIS Area-5 (%)**			**S3DIS 6-fold (%)**			**ScanNetV2 (%)**	
	OA	**mAcc**	**mIoU**	**OA**	**mAcc**	**mIoU**	**Val mIoU**	**Test mIoU**
PointNet [30]	-	48.9	41.1	78.5	66.2	47.6	-	-
PointNet++ [32]	86.4	61.2	56.0	87.3	76.2	66.7	53.5	55.7
PointCNN [20]	85.9	63.9	57.3	88.1	75.6	65.4	-	45.8
PAT [52]	-	70.8	60.1	-	76.5	64.3	-	-
PointASNL [51]	87.7	68.5	62.6	88.8	79.0	68.7	63.5	66.6
RepSurf-U [35]	90.2	76.0	68.9	90.8	82.6	74.3	-	70.0
PointNeXt-XL [33]	90.6	70.5	-	90.3	74.9	-	71.5	71.2
PointTransformer [59]	90.8	76.5	70.4	90.2	81.9	73.5	70.6	-
PointTransformer-V2 [48]	**91.1**	**77.9**	71.6	-	-	-	75.4	75.2
ADS	**91.1**	77.8	**71.8**	**91.7**	**83.7**	**75.1**	**75.6**	**75.2**

Table 3. **Semantic segmentation on area-5 and 6-fold of S3DIS**: ADS versus others. mIoU denotes the mean of per-class IoU.

Method	**mIoU (%)**	**Method**	**mIoU (%)**
SpiderCNN [50]	85.3	SyncSpecCNN [55]	81.4
PointCNN [20]	86.1	PointNet [30]	83.7
PointASNL [51]	86.1	PointCNN [20]	84.6
PointMLP [26]	86.1	RSNet [16]	84.9
RS-CNN [22]	86.2	PointNet++ [32]	85.1
PointTransformer [59]	86.6	DGCNN [44]	85.1
PointNeXt-S [33]	86.7	Tsai [40]	85.1
ADS	**86.9**	Point2Sequence [40]	85.2

Table 4. **Part segmentation on ShapeNetPart**. We report only the mean IoU and leave the result for each category in Appendix A.5.

consider a CAD-based dataset (ModelNet40 [49]) and a real-world dataset (ScanObjectNN [41]). We then evaluate ADS for part segmentation on ShapeNet [54] and semantic segmentation on the Stanford 3-D Large-Scale Indoor Spaces (S3DIS) [1]. We also conduct semantic segmentation and 3-D object detection experiments on ScanNetV2. In the ablation study, we demonstrate that ADS is less sensitive to low sampling ratios in the classification and reconstruction experiments on ModelNet40. ADS achieves competitive computation of FLOPs and parameter sizes with other attention-based methods. Experiment settings and model complexities are described in Appendixes A.1 and A.2.

4.1. 3-D Object classification

ModelNet40. The CAD-based dataset includes 12,311 objects (9,843 for training and 2,468 for testing) categorized into 40 classes. Following [32], we uniformly sample 1,024 points from each object mesh, along with the normal vectors as the input. The overall accuracy (OA) and the mean of per-class accuracy (mAcc) are adopted as evaluation metrics. Table 1 shows ADS outperforms the SOTA by 0.7% and achieves 95.1% OA without multi-scaling inference.

ScanObjectNN. To further investigate the performance of our method, we evaluate on more realistic and challenging dataset, ScanObjectNN, which comprises 2,902 real-world point clouds over 15 classes. Compared to CAD objects, the point clouds are messy (e.g., background) without alignment. We select the hardest perturbed variant (PB_T50_RS variant) for fair comparison. The OA and mAcc results in Table 2 show that ADS obtains 2.1% and 1.2% improvements in more challenging cases and manifests its general purpose. Although PointNeXt-S [33] achieves the best results, they are obtained mostly due to extended training on tuning the optimization parameters, rather than algorithmic design.

4.2. 3-D segmentation

Semantic segmentation. The evaluations are done on the S3DIS and ScanNetV2 datasets. The S3DIS consists of 271 scenes from six different areas, with a total of 13 semantic labels. To validate the performance of ADS on S3DIS, we first conduct experiments on Area 5 using three evaluation metrics: OA, mAcc, and mIoU. As shown in Table 3, compared with the SOTA method, ADS achieves comparable OA with a higher mIoU of 0.2%. To further evaluate the effectiveness of our method, we conduct experiments on other areas of S3DIS using the 6-fold cross-validation setting. The outcomes show that ADS surpasses the previous SOTA by 1.5%, 1.6%, and 1.6% on OA, mAcc, and mIoU, respectively.

Method	Backbone	ScanNetV2 (%)	
		mAP@0.25	mAP@0.5
VoxNet [27]	PointNet++ [32]	62.9	39.9
VoxNet [27]	LG3D [17]	65.1	43.0
H3DNet [58]	PointNet++ [32]	64.4	43.4
H3DNet [58]	PointNet++ (4×) [32]	67.2	48.1
3DETR [28]	Transformer	65.0	47.0
GroupFree(L12, O256) [25]	PointNet++ (w2×) [32]	69.1	52.8
GroupFree(L12, O256) [25]	LG3D [17]	70.9	54.1
GroupFree(L12, O256) [25]	RepSurf-U [35]	71.2	54.8
GroupFree(L12, O256) [25]	**ADS**	**71.8**	**55.0**

Table 5. **Detection on ScanNetV2**: ADS versus others. mAP@γ denotes the mean average precision with IoU threshold set to γ.

Furthermore, on the ScanNetV2, which comprises 1,513 indoor scenes with 20 semantic labels, ADS attains a val mIoU of 0.2% higher than the previous SOTA in comparison.

Object part segmentation. The part segmentation experiment is done on the ShapeNet dataset, which contains 16,880 models (14,006 for training and 2,874 for testing) over 16 classes. There are 50 types of the part semantic label and each object consists of between 2 to 6 parts. We follow [32] to sample each object with 2,048 points for fair comparison. Regarding the evaluation metrics, we provide the mIoU of each category and the overall mIoU. Table 4 shows that ADS yields comparable performance with PointNeXt-S [33] in overall mIoU and achieves top mIoU results for 12 categories without adopting any class-balancing loss, *e.g.*, Focal loss.

4.3. 3-D object detection

ScanNetV2. To further demonstrate the general purpose of ADS, we provide the experimental result on 3-D object detection. As mentioned in Section 3.6, the detection architecture is slightly different from the segmentation model. Following [35], we adopt Group-Free [25] as the framework and replace the backbone with ADS to obtain the detection outcomes on ScanNetV2 with 18 object classes. We report the mean Average Precision (mAP) as the standard evaluation metric under two IoU thresholds: 0.25 and 0.5 in Table 5. ADS achieves improvements (0.6% and 0.2%) from [32] and the previous SOTA [25] under the same experiment setting and the same detection framework.

4.4. Ablation study

Effect of sampling ratio. We define the *sampling ratio* as $r = m/N$. To study its effect, we assess the changes in the classification and reconstruction performances on ModelNet40 with respect to different sampling ratios, where in this case $N = 1,024$. Following [6], we set $1/r$ to be within the range of $[1, 64]$. The classification settings are the same as those in Section 4.1 except that the number of sampled points now could vary. In the reconstruction experiment, to ensure fair evaluation, we adopt the decoder in Point Completion Networks (PCN) [56] to reconstruct points from the extracted features for all methods. We use the Chamfer Distance (CD) metric to measure the inefficiency between the original and the reconstructed points. The comparison includes general frameworks (PointNet and PointNet++) and specific methods, designed explicitly for sampling, (SampleNet [19] and MetaSampler [6]). Figure 6 shows that ADS exhibits less performance degradation in both classification and reconstruction experiments, when decreasing the sampling ratio (*i.e.*, increasing $1/r$). The outcomes indicate that the sampled points by ADS are more representative than those by other techniques. Notice that here SampleNet and MetaSampler adopt PointNet as the feature extractor.

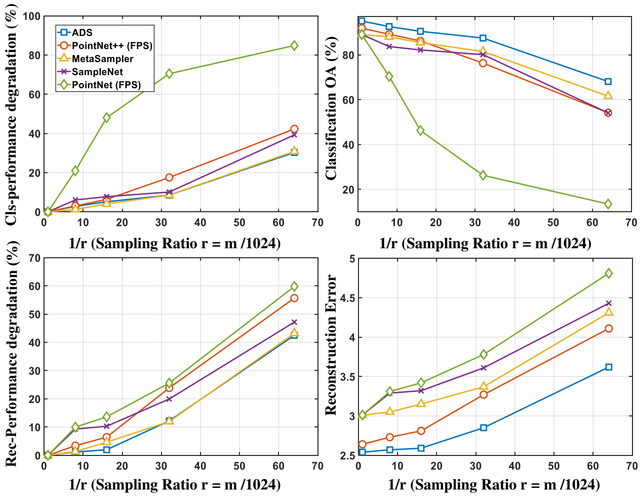

Figure 6. Classification (top row) and Reconstruction performance (bottom row) versus sampling ratio: m/N on ModelNet40 (top row). Top-left: Cls-Performance degradation (%): classification accuracy reduction in percentage; Bottom-left: Rec-Performance degradation (%): reconstruction error increase in percentage.

Effect of each component. To verify the usefulness of the various components in our method, we conduct an ablation study and report the findings in Table 6. The reconstruction term ($\mathcal{L}_{\text{EMD}}$) is crucial in that it ensures the features of the clusters are representative for reconstructing the original point set. This is one of the main reasons that the performance of ADS is more robust to the sampling ratio. In addition, the relative angle $\Delta\theta$ is another indispensable factor when calculating the self-attention relationship. The reasoning is that in 3-D space, the normal vectors of objects provide additional geometric relations beyond the relative positions from the object coordinates. Though the regularization term ($\mathcal{L}_{\text{sample}}$), the size of the cluster (n_c) and the complexity of the cluster (σ_c) are not as pivotal as the just-mentioned two components, they all positively contribute to the performance improvement. Finally, we see that our cluster-based formulation to respect the structure-aware property yields a substantial gain of 1.1% in OA.

Model	*Cluster*	$\mathcal{L}_{EMD}$	$\mathcal{L}_{sample}$	n_c	$\Delta\theta$	σ_c	**OA(%)**
I	✓	✓					93.9
II	✓	✓		✓			94.1
III	✓	✓		✓	✓		94.5
IV	✓	✓		✓	✓	✓	94.8
V	✓			✓	✓	✓	93.7
VI	✓		✓	✓	✓	✓	94.0
VII	✓	✓	✓	✓	✓	✓	**95.1**
VIII		✓	✓	✓	✓	✓	94.0

Table 6. Ablation of the key components in our method.

Classification (Model40)			
k	**Scaling Ratio**	Δ	**OA (%)**
8	[×1.0, ×2.0, ×2.0]	-0.6	94.5
8	[×1.0, ×2.0, ×4.0]	-1.5	93.6
16	[×1.0, ×2.0, ×2.0]	0	**95.1**
16	[×1.0, ×2.0, ×4.0]	-0.8	94.3

Segmentation (S3DIS)			
k	**Scaling Ratio**	Δ	**mIoU (%)**
8	[×1.0, ×2.0, ×2.0, ×2.0]	0	**71.8**
8	[×1.0, ×2.0, ×2.0, ×4.0]	-1.5	70.3
16	[×1.0, ×2.0, ×2.0, ×2.0]	-1.9	69.9
16	[×1.0, ×2.0, ×4.0, ×4.0]	-0.7	71.1

Table 7. Ablation of different k values and scaling ratios.

Model design: k-point neighborhood. The parameter k in $\mathcal{N}_k(\mathbf{p}_{o_c})$ is pivotal to ADS cluster-wise sampling and affects the outcome of the targeted downstream task. Specifically, we investigate the impact of k on the classification and segmentation tasks and report the findings in Table 7. In our formulation, similar to PointNet++ with the FPS scheme, the k value for each ADS module is determined via multiplying the initial number by the scaling ratio. For example, configuration 8 with [×1.0, ×2.0, ×2.0] indicates that the k values in the three ADS modules are specified by [8, 16, 32] in the classification architecture. We empirically find that the optimal values of k for both classification and segmentation tasks are 16, 32, and 64, and 4, 8, 32, and 64, respectively. These results demonstrate the significance of appropriately selecting the k value to achieve good performance for ADS.

Positional encoding. When the self-attention operation does not take account of the information of relative position, it is advantageous to consider positional encoding. Vaswani *et al*. [42] consider a mapping function to encode the position information in self-attention, while other approaches, *e.g*., [4, 14, 15, 24], embrace the strategy of using a lookup table with learnable parameters to better model the position information. Similar to [23] we apply MLP to predict the relative factors relevant to the underlying 3-D task. Table 8 shows that the proposed MLP formulation of positional encoding (with relative position Δxyz and normal vector $\Delta\theta$) better explores the geometric relations in a point cloud.

Positional encoding type	**OA (%)**
None	93.9
Fixed mapping function (cos,sin)	94.0
Look-up table for learnable value (Relative: Δxyz)	94.0
MLP prediction (Relative: Δxyz)	94.1
MLP prediction (Relative: $\Delta xyz + \Delta\theta$)	**94.4**

Table 8. Ablation of different positional encoding schemes.

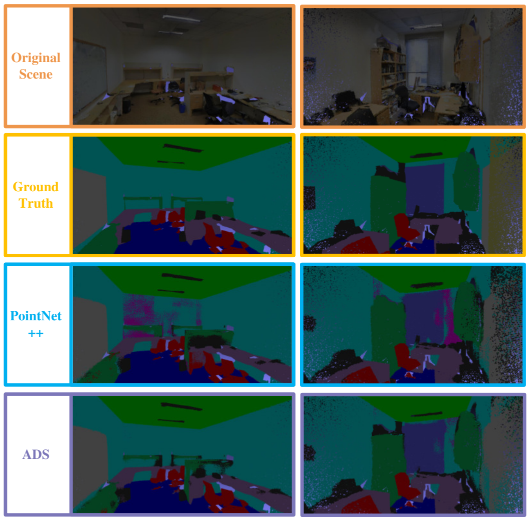

Figure 7. **Qualitative comparison** on Area 5 of S3DIS. Examples of dense semantic segmentation by ADS and (baseline) PointNet++.

Qualitative comparison. Figure 7 visually illustrates an example of dense semantic segmentation from Area 5 of S3DIS. The segmentation outcomes by ADS reveal more details about the scene than those by the standard PointNet++. For example, ADS successfully segments the whole scene; in contrast, the standard PointNet++ misses the wall region.

5. Conclusions

Driven by the cluster-based design, the proposed *attention discriminant sampling* (ADS) implicitly achieves structure-aware sampling by taking account of not only geometric relations but also semantic relations in point clouds. Its effectiveness for deciding how the sampled points are cluster-wise distributed results from exploring the within-cluster and between-cluster self-attention responses that quantitatively measure each cluster's complexity and similarities to others. We further consolidate the learning of ADS with an EMD regularization loss to ensure the sampled points are representative. With its good performance, ADS provides a general solution for 3-D point cloud sampling and applications.

Acknowledgements. This work was supported in part by NSTC grants 111-2221-E-001-011-MY2, 111-2221-E-001-015-MY3 and 111-2634-F-007-010 of Taiwan. We are grateful to National Center for High-performance Computing for providing computational resources and facilities.

References

[1] Iro Armeni, Ozan Sener, Amir Roshan Zamir, Helen Jiang, Ioannis K. Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1534–1543. IEEE Computer Society, 2016.

[2] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Trans. Graph.*, 37(4):71, 2018.

[3] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022.

[4] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. PMLR, 2020.

[5] Chen Chen, Zhe Chen, Jing Zhang, and Dacheng Tao. SASA: semantics-augmented set abstraction for point-based 3d object detection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 221–229. AAAI Press, 2022.

[6] Ta Ying Cheng, Qingyong Hu, Qian Xie, Niki Trigoni, and Andrew Markham. Meta-sampler: Almost-universal yet task-oriented sampling for point clouds. *CoRR*, abs/2203.16001, 2022.

[7] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[8] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, volume 9912 of *Lecture Notes in Computer Science*, pages 628–644. Springer, 2016.

[9] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2760–2769. Computer Vision Foundation / IEEE, 2019.

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2463–2471. IEEE Computer Society, 2017.

[11] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3809–3820. PMLR, 2021.

[12] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. MVTN: multi-view transformation network for 3d shape recognition. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 1–11. IEEE, 2021.

[13] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. *CoRR*, abs/2203.10314, 2022.

[14] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3588–3597. Computer Vision Foundation / IEEE Computer Society, 2018.

[15] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3463–3472. IEEE, 2019.

[16] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2626–2635. Computer Vision Foundation / IEEE Computer Society, 2018.

[17] Yaomin Huang, Xinmei Liu, Yichen Zhu, Zhiyuan Xu, Chaomin Shen, Zhengping Che, Guixu Zhang, Yaxin Peng, Feifei Feng, and Jian Tang. Label-guided auxiliary training improves 3d object detector. *CoRR*, abs/2207.11753, 2022.

[18] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022.

[19] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 7575–7585. Computer Vision Foundation / IEEE, 2020.

[20] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 828–838, 2018.

[21] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[22] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8895–8904. Computer Vision Foundation / IEEE, 2019.

[23] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer V2: scaling up capacity and resolution. *CoRR*, abs/2111.09883, 2021.

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9992–10002. IEEE, 2021.

[25] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2929–2938. IEEE, 2021.

[26] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[27] Daniel Maturana and Sebastian A. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, pages 922–928. IEEE, 2015.

[28] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2886–2897. IEEE, 2021.

[29] Carsten Moenning and Neil A. Dodgson. Fast marching farthest point sampling. In Julián Flores and Pedro Cano, editors, *24th Annual Conference of the European Association for Computer Graphics, Eurographics 2003 - Posters, Granada, Spain, September 1-5, 2003*. Eurographics Association, 2003.

[30] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85. IEEE Computer Society, 2017.

[31] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5648–5656. IEEE Computer Society, 2016.

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5099–5108, 2017.

[33] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *CoRR*, abs/2206.04670, 2022.

[34] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11143–11152, 2022.

[35] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18920–18930. IEEE, 2022.

[36] Haoxi Ran, Wei Zhuo, Jun Liu, and Li Lu. Learning inner-group relations on point clouds. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15457–15467. IEEE, 2021.

[37] Riccardo Roveri, Lukas Rahmann, Cengiz Öztireli, and Markus H. Gross. A network architecture for point cloud classification via automatic depth images generation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4176–4184. Computer Vision Foundation / IEEE Computer Society, 2018.

[38] Kripasindhu Sarkar, Basavaraj Hampiholi, Kiran Varanasi, and Didier Stricker. Learning 3d shapes as multi-layered height-maps using 2d convolutional networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, volume 11220 of *Lecture Notes in Computer Science*, pages 74–89. Springer, 2018.

[39] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 945–953. IEEE Computer Society, 2015.

[40] Meng-Shiun Tsai, Pei-Ze Chiang, Yi-Hsuan Tsai, and Wei-Chen Chiu. Self-supervised feature learning from partial point clouds via pose disentanglement. *CoRR*, abs/2201.03018, 2022.

[41] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1588–1597. IEEE, 2019.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon,

Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[43] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.*, 36(4):72:1–72:11, 2017.

[44] 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):146:1–146:12, 2019.

[45] Yikai Wang, TengQi Ye, Lele Cao, Wenbing Huang, Fuchun Sun, Fengxiang He, and Dacheng Tao. Bridged transformer for vision and point cloud 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12114–12123, 2022.

[46] Ziyi Wang, Yongming Rao, Xumin Yu, Jie Zhou, and Jiwen Lu. Semaffinet: Semantic-affine transformation for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11819–11829, 2022.

[47] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9621–9630. Computer Vision Foundation / IEEE, 2019.

[48] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer V2: grouped vector attention and partition-based pooling. *CoRR*, abs/2210.05666, 2022.

[49] 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 Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1912–1920. IEEE Computer Society, 2015.

[50] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212 of *Lecture Notes in Computer Science*, pages 90–105. Springer, 2018.

[51] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 5588–5597. Computer Vision Foundation / IEEE, 2020.

[52] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3323–3332. Computer Vision Foundation / IEEE, 2019.

[53] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6686, 2022.

[54] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 35(6):210:1–210:12, 2016.

[55] Li Yi, Hao Su, Xingwen Guo, and Leonidas J. Guibas. Syncspeccnn: Synchronized spectral CNN for 3d shape segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6584–6592. IEEE Computer Society, 2017.

[56] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: point completion network. In *2018 International Conference on 3D Vision, 3DV 2018, Verona, Italy, September 5-8, 2018*, pages 728–737. IEEE Computer Society, 2018.

[57] Yanan Zhang, Jiaxin Chen, and Di Huang. Cat-det: Contrastively augmented transformer for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 908–917, 2022.

[58] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XII*, volume 12357 of *Lecture Notes in Computer Science*, pages 311–329. Springer, 2020.

[59] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 16239–16248. IEEE, 2021.