Perturbed Self-Distillation: Weakly Supervised Large-Scale Point Cloud Semantic Segmentation

Yachao Zhang[1], Yanyun Qu[1*], Yuan Xie[2*], Zonghao Li[1], Shanshan Zheng[1], Cuihua Li[1]
[1] Xiamen University, [2] East China Normal University
yachaozhang@stu.xmu.edu.cn, yyqu@xmu.edu.cn, yxie@cs.ecnu.edu.cn,
{zonghaoli,shanshanzheng}@stu.xmu.edu.cn, chli@xmu.edu.cn

Abstract

Large-scale point cloud semantic segmentation has wide applications. Current popular researches mainly focus on fully supervised learning which demands expensive and tedious manual point-wise annotation. Weakly supervised learning is an alternative way to avoid this exhausting annotation. However, for large-scale point clouds with few labeled points, the network is difficult to extract discriminative features for unlabeled points, as well as the regularization of topology between labeled and unlabeled points is usually ignored, resulting in incorrect segmentation results.

To address this problem, we propose a perturbed self-distillation (PSD) framework. Specifically, inspired by self-supervised learning, we construct the perturbed branch and enforce the predictive consistency among the perturbed branch and original branch. In this way, the graph topology of the whole point cloud can be effectively established by the introduced auxiliary supervision, such that the information propagation between the labeled and unlabeled points will be realized. Besides point-level supervision, we present a well-integrated context-aware module to explicitly regularize the affinity correlation of labeled points. Therefore, the graph topology of the point cloud can be further refined. The experimental results evaluated on three large-scale datasets show the large gain (3.0% on average) against recent weakly supervised methods and comparable results to some fully supervised methods.

1. Introduction

Currently, large-scale point cloud semantic segmentation attracts more and more attention due to its broad applications in environmental perception, such as autonomous driving, human-computer interaction, virtual reality, and robotics. Great progress has been made in small-scale point cloud semantic segmentation [15, 16, 31, 12, 11, 22, 24].

*Corresponding Author

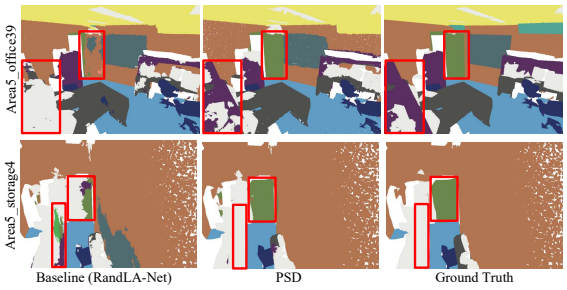


Figure 1. Semantic segmentation results with 1% labeled points. We improve the segmentation accuracy of the categories with high structural similarity to other categories highlighted by the red box.

Recently, RandLA-Net [5] was proposed as an efficient method for large-scale point cloud ($\sim 10^6$ points) semantic segmentation. However, the mainstream of all these methods is built upon fully supervised learning, requiring tremendous point-wise annotation. Unfortunately, such annotation involves a large amount of manual work. For example, it takes 22.3 minutes to annotate a scene of ScanNet [2] on average [23].

To avoid the exhausting annotation, weakly supervised methods are rising. Xu and Lee *et al.* [26] firstly proposed a weakly supervised method by labeling a tiny fraction points. This method utilizes multi-branch supervision and a parameter-free graph-based smoothing item based on Laplacian matrix, achieving comparable performance to its fully supervised version with $10\times$ fewer labeled points. However, this method is limited as *it can not directly be implemented to the large-scale point cloud with fewer labels*, due to the lack of learnable topological relationship and the high computational complexity of the Laplacian matrix. In addition, this method only uses point-level supervision, and it is not easy to model the context. While in fully supervised segmentation task, context information is implicitly learned by U-Net-style structure [16, 5] or local feature aggregation [31, 22]. Due to the limited annotation of large-scale scenarios, these techniques cannot meet the requirements

for learning enough discriminative features. For example, in the first column of Figure 1, the Baseline (RandLA-Net [5]), trained by 1% labels, misclassifies many points in the red box.

Inspired by the successes of self-supervised learning, we propose a perturbed self-distillation framework that focuses on solving two critical issues: 1) How to design auxiliary supervision for unlabeled points, such that a well-formed point graph topology can be established. 2) Besides merely supervising on point-level, how to derive a context regularization for modeling the relationship among labeled points?

For the first issue, we introduce the perturbed self-distillation by constructing a perturbed branch and keeping the predicted distribution consistency between the perturbed branch and the original branch. The consistency constraints provide additional supervision for all points, enabling the introduced graph convolutional networks (GCNs) to establish a well-formed graph topology among all points. Therefore, based on this learned graph structure, a new interactive way of two branches is introduced, which realize the effective information flow between labeled and unlabeled points.

Secondly, to refine the graph topology, we propose the context-aware module, where we encode the semantic correlation affinity of the labeled points to supervise the learning of feature correlation. Since the labeled points are distributed in the graph topology like anchor points, if the relationship of the anchor points could be guaranteed to be correct to some extent, it would have a positive impact on the classification results of unlabeled data.

To summarize, our contributions are three-fold:

• We propose a perturbed self-distillation (PSD) framework, where a self-distillation mechanism is introduced by constructing perturbed samples to ensure the predictive consistency among perturbed samples and original samples. Accompanying with the supervision from labeled data, the graph topology of the whole point cloud can be effectively established through the information propagation between labeled and unlabeled points during training.

• A context-aware module is presented, and it can be seamlessly integrated into the self-distillation framework. With the help of exactly learning the affinity context of labeled points, the graph topology of the point cloud can be further refined.

• PSD achieves significant performance over state-of-the-art methods and gains a 3.0% improvement on average of three datasets. Moreover, PSD also improves the performance of Baseline in the way of fully supervised learning.

2. Related work

2.1. Large-scale point cloud segmentation

On the strength of the prominent advances of deep neural networks after PointNet and PointNet++ [15, 16], semantic segmentation has attracted more attention. Though some works [31, 12, 11, 22, 24, 9, 6, 28] have shown promising results, most of them only work on small point clouds and cannot directly scale up to large-scale point clouds due to high computational costs or memory requirements [5].

Recently, the graph convolution-based method SPGraph [8] and voxel-based method FCPN [17] are proposed for large-scale point clouds analysis. However, SPGraph or voxelization is computationally expensive. RandLA-Net [5] utilizes a random point sampling strategy instead of more complex point selection approaches, which provides an efficient and lightweight neural architecture. These state-of-the-art methods mentioned above all depend on well-labeled point cloud datasets. However, this point-wise annotation is labor-intensive and time-consuming.

2.2. Weakly supervised point cloud segmentation

The study of weakly supervised point cloud semantic segmentation is in its infancy. We divide the existing methods into three categories according to the manner of annotation: a tiny fraction of points annotated methods [26, 30], semantic category annotated method [23], and 2D segmentation maps annotated method [21].

Annotating a tiny fraction of points is a popular type of weakly supervised point cloud segmentation. Xu and Lee [26] utilizes multi-branch supervision and a subsequent smooth branch to ensure a better representation for the small-scale point cloud. A parameter-free graph used for post-processing is not learnable and will cause GPU memory explosion at handling large-scale point clouds. Zhang *et al.* [30] proposed a transfer learning-based method to improve the performance of weakly supervised point cloud segmentation. This method requires additional datasets to learn prior knowledge and transfers the knowledge to weakly supervised segmentation tasks. However, the pre-training is time-consuming and requires abundant data.

Unlike the multi-branch constraints of Xu and Lee [26], we focus on implicit label propagation by constructing the learnable graph topology, and our method is suitable to large-scale point clouds. Different from Zhang *et al.* [30], the needs for additional datasets for pre-training, our method focuses on how to mine the supervision information of the data itself.

In addition to the above methods, MPRM [23] utilizes the semantic categories of sub-cloud and introduces a point class activation map (PCAM) using a classification network. It mines the localization cues for each class from various aspect features to generate pseudo point-level labels. GPFN [21] uses a deep graph convolutional network-based framework and leverages 2D segmentation maps of different viewpoints to supervise the point cloud training.

However, the two methods need to split the point cloud into sub-clouds or truncated point clouds that inevitably

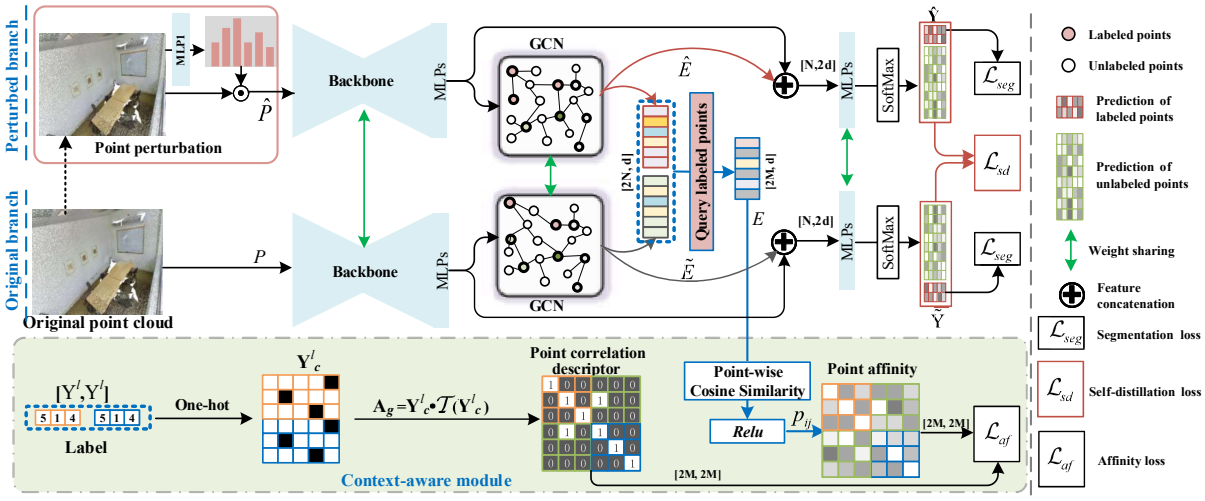


Figure 2. The framework of perturbed self-distillation.

lose the structural information and difficult to model the contexts. Our PSD introduces a context-aware module to refine the context of neighbor correlation.

3. Method

3.1. Notation and representation

Let P be a point cloud in the training set defined as $\{(X^l, Y^l), (X^u, \varnothing)\} = \left\{\left(x_1^l, y_1^l\right), \ldots, \left(x_M^l, y_M^l\right), x_{M+1}^u, \ldots, x_N^u\right\}$, where X^l and X^u are sets of M labeled points and the unlabeled points respectively, and Y^l is label set of labeled points. Formally, given a large-scale point cloud with a tiny fraction of labels as input, the weakly-supervised segmentation aims to learn the function $f : X^l \cup X^u \mapsto Y$. Specifically, for 1% setting, the number of labeled points is $M = 1\% \times N$. The 1pt represents only one point labeled with the ground truth of every category, and the number of labeled points M equals the number of categories C. All the labeled points are selected randomly.

In PSD framework, there are two networks: the backbone and the graph convolutional network (GCN). For the backbone and GCN, we choose RandLA-Net [5] and EdgeConv [22], where the GCN can be formulated as:

$$\mathcal{E}_{gcn}(x_i^k) = \mathcal{F}(\{x_i \oplus \left(x_i - x_i^k\right) | x_i^k \in \mathcal{N}\left(x_i\right)\}; \Theta), \quad (1)$$

where $\mathcal{N}\left(x_i\right)$ is the local neighborhood of point x_i simply constructed by the K nearest neighbors algorithm based on the Euclidean distances, $\oplus$ denotes feature concatenation. $\mathcal{F}$ is a function with a set of learnable parameters Θ. Then we use the $Max\text{-}pooling$ operation to aggregate local features. Thus GCN can be formulated as:

$$\mathcal{G}(x_i; \Theta) = \max_{x_i^k \in \mathcal{N}(x_i)} \mathcal{E}_{gcn}(x_i^k). \quad (2)$$

3.2. Overall framework

For weakly supervised tasks, we consider two ways to make the robust feature representation for point cloud segmentation: 1) introducing auxiliary supervision to construct graph topology for information flow, 2) and refining the graph topology. We propose a perturbed self-distillation framework shown in Figure 2, containing perturbed self-distillation (top part) and a pluggable context-aware module (bottom part). The former constructs a perturbed branch and constraints the predictive consistency among perturbed samples and original samples. As a result, the GCN can effectively establish graph topology of the whole point by the supervision from labeled data and consistency constraint. Besides, the latter is designed to learn the exact affinity context of labeled points, such that the graph topology of the point cloud can be further refined.

In Figure 2, a training batch is firstly fed into the perturbed branch and the original branch, respectively. In the perturbed branch, point clouds need to be disturbed by two random transformations and a learnable transformation. Original point clouds and perturbed point clouds pass through the backbone and GCN layers. Then, we concatenate the output features of backbone and GCN to refine the prediction. The output category probability distributions of two branches are supervised by cross-entropy loss for labeled points and self-distillation loss for all points. Moreover, the context-aware module constructs a point correlation descriptor to constrain point-wise feature affinity through an affinity loss.

3.3. Perturbed self-distillation

Perturbed self-distillation is very suitable for weakly supervised semantic segmentation for two reasons: (1) Weak supervision tasks benefit from the additional supervision

created by self-distillation. (2) Self-distillation can transfer knowledge between different perturbed branches [25] and drive the network to automatically learn more representative feature for generalization. However, to implement the perturbed self-distillation, two difficulties appear: (1) How to construct the perturbed branch. (2) How to conduct interaction between two branches. We will detail the solution as follows.

3.3.1 Perturbed branch

The two branches respectively take a point cloud and the corresponding perturbed point cloud as inputs, and output the class probability distributions. Since point perturbation is crucial for self-distillation learning, a strong perturbation may cause the network difficult to converge, while a to weak perturbation will make the performance trivial. Thus, we design a combinational transformation which contains the scene-wise transformation, the point-wise displacement for coordinates, and the attribute attention. Note that the perturbed branch will not be used during the test stage.

Scene-wise transformation. A point cloud P can be split to the coordinate $P_x \in \mathbb{R}^{N\times 3}$ and the attribute P_a (*e.g.,* color or normal). We use the coordinate for scene-wise transformation which contains random rotating $T^r \in \mathbb{R}^{3\times 3}$ and mirroring $T^m \in \mathbb{R}^{3\times 3}$. The rotated point cloud P_x^r can be denoted as $P_x^r = P_x \cdot T^r$, where $T^r = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$, the rotation degree θ around the z-axis obeys a uniformly distribution $U(0, 2\pi)$, "$\cdot$" denotes the matrix product. For mirroring, we only consider the mirror transformation relative to the Y-axis as $P_x^m = P_x \cdot T^m$ and $T^m = \mathrm{diag}\,(1, -1, 1)$.

Point-wise displacement. For the point-wise displacement, we jitter the point location and yield a noise displacement $T^j \in \mathbb{R}^{N\times 3}$, where T^j is the Gaussian noise with mean 0.01 and variance 1.0. Then, the offset is set in $[-0.05, 0.05]$. The jittered point cloud denotes $P_x^j = P_x + T^j$. The point cloud with perturbed coordinate $\hat{P}_x$ can be randomly selected from $\{P_x^r, P_x^m, P_x^j\}$.

Attribute attention. For different point clouds, their attributes usually make different influence on the discriminability of extracted features. The real-world point cloud has diversity with different attributes. For example, the color attribute plays a critical role in some categories for obtaining discriminative features, such as the "door" and "window". However, in some categories with high color similarity (*e.g.,* the "column" and "wall"), the color information may confuse feature extraction.

The above two random transformation methods can not handle the difference in attributes. Therefore, we introduce an attribute attention layer to adaptively learn the weights for input attributes, which can be regarded as a learnable transformation to fit the diversity of point clouds. Specifically, we concatenate the perturbed coordinate $\hat{P}_x$ and original attribute P_a as P_c. Then, a mapping function $\mathcal{F}_a(\cdot, \Theta_a)$ is implemented by a multi-layer perceptron with learnable parameters Θ_a to map the channel of P_c to the response $s = \mathcal{F}_{\mathrm{a}}\left(P_c; \Theta_a\right), \in \mathbb{R}^{N\times d}$. The attribute attention score α_i of the channel i can be formulated as:

$$\alpha_i = \frac{\exp\left(s_i\right)}{\sum_{i=1}^{d} \exp\left(s_i\right)}. \tag{3}$$

We use attribute attention scores to construct a diagonal matrix $\boldsymbol{\alpha} = \mathrm{diag}\,(\alpha_1, \alpha_2, \cdots, \alpha_d)$. The final perturbed point cloud can be obtained by $\hat{P} = P_c \cdot \boldsymbol{\alpha}$.

3.3.2 Self-distillation loss

To enforce the constraints of the predictive consistency, we use the Jensen-Shannon divergence as the self-distillation loss $\mathcal{L}_{sd}$ to constrain the distribution of class probabilities between the original branch and the perturbed branch:

$$\begin{aligned} \mathcal{L}_{sd} &= \frac{1}{2N}\sum_{i=1}^{N} JS(\tilde{y}_i \| \hat{y}_i) \\ &= \tilde{y}_i \log\left(\frac{2\tilde{y}_i}{\tilde{y}_i + \hat{y}_i}\right) + \hat{y}_i \log\left(\frac{2\hat{y}_i}{\tilde{y}_i + \hat{y}_i}\right), \end{aligned} \tag{4}$$

where $\tilde{y}_i$ and $\hat{y}_i$ are the predicted probabilities of point i output by the original branch and the perturbed branch, respectively.

The self-distillation loss does not emphasize whether the two branches can predict the category accurately, but focus on ensuring the predictive consistency between two branches, leading to an auxiliary loss function in the scenario of self-supervision. With the help of the self-distillation loss and cross-entropy loss for labeled points, the accurate graph topology of the whole point cloud can be achieved through information propagation among labeled and unlabeled data.

3.4. Context-aware module

The architecture of U-Net consisted of a contracting path trying to capture contexts [18], which is widely used in point cloud understanding tasks [31, 5]. As there are few labeled points, such an implicit context is insufficient to understand large-scale, irregular, and disordered point clouds. Therefore, we present a context-aware module to model the exact affinity context of labeled points. The labeled points are distributed in the graph topology like anchor points. The more accurate the correlation between anchor points in graph topology is, the more precise the prediction of unlabeled points becomes. The context-aware module comprises three essential components: the point affinity, the point semantic correlation descriptor, and an affinity loss.

Point affinity. We introduce the point affinity matrix to denote the similarity of point features. Let $\tilde{E} \in \mathbb{R}^{N\times d}$ and $\hat{E} \in \mathbb{R}^{N\times d}$ be the output features of the GCN layer in two branches, respectively. We concatenate $\tilde{E}$ and $\hat{E}$ along point dimension and query the merged features of labeled points as $E = \{e_1, e_2, \ldots, e_{2M}\}$, where e_i is the feature vector of point i. The point affinity is constructed by the cosine similarity:

$$p_{ij} = max(0, \frac{\langle e_i, e_j \rangle}{\|e_i\|\|e_j\|}), i, j = 1, 2, \ldots 2M, \quad (5)$$

where $\|\cdot\|$ denotes ℓ_2 normalization, "$\langle\cdot,\cdot\rangle$" denotes inner product. Obviously, if two points belong to different categories, the p_{ij} is small. On the contrary, it will be larger.

Point correlation descriptor. We design the point correlation descriptor to provide supervision for point affinity. It indicates the pairwise relationship of labeled samples in a point cloud. Specifically, we first convert the labels Y^l into the one-hot vector. As the perturbed branch and the original branch have the same label, we merge the two one-hot codes $Y_c^l = [Y_h^l; Y_h^l] \in \mathbb{R}^{2M\times C}$, where M and C are the numbers of labeled points and categories. This descriptor of semantic correlation can be defined as:

$$A_g = Y_c^l \cdot \mathcal{T}(Y_c^l), \ \in \mathbb{R}^{2M\times 2M} \quad (6)$$

where $\mathcal{T}(\cdot)$ is the matrix transpose. Let $a_{ij} \in \{0, 1\}$ denote an element of A_g, it represents the semantic correlation of point i and j. For example, $a_{ij} = 1$ indicates that i and j belong to the same category.

Affinity loss. Inspired by [29], we use a compound loss, which consists of the cross-entropy loss $\mathcal{L}_{ce}$, precision $\mathcal{L}_p$, and recall $\mathcal{L}_r$, to guide the learning of the point affinity:

$$\mathcal{L}_{af} = \mathcal{L}_{ce} + \mathcal{L}_p + \mathcal{L}_r. \quad (7)$$

The cross-entropy loss $\mathcal{L}_{ce}$ can be formulated as:

$$\mathcal{L}_{ce} = -\frac{1}{2(2M)^2} \sum_{i=1}^{(2M)} \sum_{j=1}^{(2M)} (a_{ij} \log p_{ij} + (1-a_{ij}) \log(1-p_{ij})), \quad (8)$$

where p_{ij} is the element of feature affinity at the point i and j. The precision and recall can be separately formulated as:

$$\mathcal{L}_p = -\frac{1}{2M} \sum_{j=1}^{2M} \big(\log \frac{\sum_{i=1}^{2M} a_{ij} p_{ij}}{\sum_{i=1}^{2M} p_{ij}} \big), \quad (9)$$

$$\mathcal{L}_r = -\frac{1}{2M} \sum_{j=1}^{2M} \big(\log \frac{\sum_{i=1}^{2M} a_{ij} p_{ij}}{\sum_{i=1}^{2M} a_{ij}} \big). \quad (10)$$

Discussion. The differences between the context prior layer (CPL) [29] and our context-aware module appear in the following aspects. 1) The task is different. Ours is used for weakly supervised point cloud segmentation, while CPL is employed for fully supervised 2D scene segmentation. 2) The objective is different. We build point affinity to optimize the learning of the graph topology to facilitate the information flowing between labeled and unlabeled points instead of learning the intra-class and inter-class contextual dependencies in CPL. Moreover, we combine the two branches together to calculate the point affinity matrix, which provides the consistency constraint of feature-level for the self-distillation.

3.5. Total loss

The total loss $\mathcal{L}$ contains three terms: the segmentation loss $\mathcal{L}_{seg}$, the self-distillation loss $\mathcal{L}_{sd}$, and the affinity loss $\mathcal{L}_{af}$.

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{sd} + \mathcal{L}_{af}, \quad (11)$$

where $\mathcal{L}_{sd}$ is given in Eq. (4) and $\mathcal{L}_{af}$ is given in Eq. (7). We utilize the softmax cross-entropy loss of labeled points as the segmentation loss for two branches. In the original branch, it can be formulated as:

$$\mathcal{L}_{seg} = -\frac{1}{CM} \sum_{i=1}^{M} \sum_{c=1}^{C} y_{ic}^l \log \frac{\exp(\tilde{y}_{ic}^l)}{\sum_{c=1}^{C} \exp(\tilde{y}_{ic}^l)}, \quad (12)$$

where y_{ic}^l denotes the ground truth label, $\tilde{y}_{ic}^l$ is the predictions of the labeled point i. M and C denote the number of labeled points and categories, respectively. The loss of the perturbed branch is the same as original branch.

4. Experiments

4.1. Experiment setting

Dataset. To show the versatility of PSD, we evaluate the performance of PSD on three large-scale datasets: S3DIS[1], ScanNet-v2 [2], and Semantic3D [3]. **S3DIS** contains 6 large-scale indoor areas including 271 rooms and each room contains about 10^6 points. We use 6 attributes (*i.e.,* XYZ coordinates and RGB colors) as the input of each point. **ScanNet-v2** is a large-scale point cloud dataset which comes from an RGB-D video containing 2.5 million views in more than 1500 scans. It provides point clouds containing RGB attributes and well-annotated points. **Semantic3D** is an outdoor dataset which provides a large labeled 3D point cloud of natural scenes with over 4 billion points in total. It covers a range of diverse urban scenes. The raw 3D points belong to 8 classes and contain 3D coordinates, RGB information, and intensity. We use the coordinate and corresponding color channels in our experiments.

Implementation details. Similar to the weakly supervised semantic segmentation method [30], we choose an efficient RandLA-Net [5] as our backbone. We use Adam Optimizer with an initial learning rate of 0.001 and momentum of 0.9 to train 100 epochs for all datasets on an

Setting	Method	mIoU	ceil.	floor	wall	beam	col.	win.	door	chair	table	book.	sofa	board	clutter
1pt (0.2%)	Π Model [7]	44.3	89.1	97.0	71.5	0.0	3.6	43.2	27.4	62.1	63.1	14.7	43.7	24.0	36.7
	MT [19]	44.4	88.9	96.8	70.1	0.1	3.0	44.3	28.8	63.6	63.7	15.5	43.7	23.0	35.8
	Xu and Lee [26]	44.5	90.1	97.1	71.9	0.0	1.9	47.2	29.3	62.9	64.0	15.9	42.2	18.9	37.5
1pt (0.03%)	Baseline	40.7	83.7	90.7	61.2	0.0	11.9	40.8	15.2	52.0	51.7	14.9	50.5	25.3	31.8
	PSD	48.2	87.9	96.0	62.1	0.0	20.6	49.3	40.9	55.1	61.9	43.9	50.7	27.3	31.1
10%	Xu and Lee [26]	48.0	90.9	97.3	74.8	0.0	8.4	49.3	27.3	69.0	71.7	16.5	53.2	23.3	42.8
1%	Zhang *et al.* [30]	61.8	91.5	96.9	80.6	0.0	18.2	58.1	47.2	75.8	85.7	65.2	68.9	65.0	50.2
	PSD	63.5	92.3	97.7	80.7	0.0	27.8	56.2	62.5	78.7	84.1	63.1	70.4	58.9	53.2
Fully	PointNet [15]	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	58.9	52.6	5.9	40.3	26.4	33.2
	SPH3D [10]	59.5	93.3	97.1	81.1	0.0	33.2	45.8	43.8	79.7	86.9	33.2	71.5	54.1	53.7
	KPConv rigid [20]	65.4	92.6	97.3	81.4	0.0	16.5	54.5	69.5	80.2	90.1	66.4	74.6	63.7	58.1
	RandLA-Net* [5]	63.0	92.4	96.7	80.6	0.0	18.3	61.3	43.3	77.2	85.2	71.5	71.0	69.2	52.3
	PSD	65.1	92.3	97.1	80.7	0.0	32.4	55.5	68.1	78.9	86.8	71.1	70.6	59.0	53.0

Table 1. Quantitative results on Area-5 of S3DIS [1]. "*" denotes the results of the method trained by us using the official code. Baseline is RandLA-Net [5]. Note that our 1pt denotes only one labeled point for each category in the entire rooms instead of small blocks (*e.g.,* 1×1 meter) of Xu and Lee [26]. The number of labeled points in our 1pt setting accounts for 0.03% of the total points, which is about 0.2% in Xu and Lee [26].

NVIDIA RTX Titan GPU. Furthermore, we adopt the mean IoU (mIoU, %) as the standard metrics. We experimentally study two types of the weak label: 1pt and 1% settings. Moreover, we further extend PSD to the fully supervised manner and conduct tests on Area-5 of S3DIS.

4.2. Experiment results

Results on Area-5 of S3DIS. In Table 1, we show the quantitative results on Area-5 of S3DIS. From the comparison of weakly supervised tasks, it is observed that PSD achieves a great improvement in the weakly settings of 1pt and 1%, respectively. At the 1pt setting, although PSD has a lower labeling rate than Xu and Lee [26], PSD still gains about a 3.7% improvement in mIoU and achieves about a 7.5% improvement comparing to the RandLA-Net with the weakly-supervised setting (Baseline). Notably, PSD with 1% labeled points gains 18.7%, 11.6% and 28.0% improvements in categories "column" (col.), "door", and "bookcase" (book.) against Xu and Lee [26], respectively. These categories are usually similar in color or structure to the "wall". These results support the argument that PSD can make the features more discriminative.

At the 1% setting, PSD outperforms a 1.7% improvement over Zhang *et al.* [30]. Besides, PSD achieves 63.5% mIoU with the gain of 15.5% against Xu and Lee with 10% labeled points. It is interesting that the performance of PSD is better than fully supervised RandLA-Net [5] by only 1% annotated points. One reason why PSD performs strongly is that PSD can get the more discriminative features by self-distillation. We give some qualitative results in Figure 1. Compared with the baseline, we can see that PSD gets the correct segmentation results, but the Baseline misclassifies the points in the red box. Because the categories in the red box are similar in structure to other categories.

To verify the scalability of PSD, we further extend PSD to fully supervised tasks. Since the large-scale point cloud contains $\sim 10^6$ points, the affinity matrix is very large and leads to GPU memory limitation at the training phase. We insert our context-aware module after the encoder with $\sim 10^2$ points based on the following considerations: 1) The point features of the current layer can competently represent a local feature. 2) The learned contextual information can be propagated to higher resolution layers by the decoder of backbone. In Table 1, it can be seen from the comparison of fully supervised settings (Fully) that we achieve a 2.1% improvement over RandLA-Net [5] and the performance is close to that of KPConv rigid [20], which demonstrates the good scalability of PSD.

Results on 6-fold of S3DIS. In Table 2, we list the quantitative results of 6-Area cross-validation (6-fold) on S3DIS. We first notice that PSD achieves the 68.0% mIoU, which indeed is superior to over Zhang *et al.* [30]. PSD exceeds the performance of fully supervised methods (*e.g.* PointCNN [13], DGCNN [22], and ShellNet [31]), and achieves comparable results to the state-of-the-art fully supervised methods.

Results on ScanNet-v2. MPRM [23] annotates the semantic categories of the subcloud-level (sub.). It reduces the labor compared to our 1% setting. But it needs to divide the entire scene into subclouds and repeatedly annotate every subcloud. From Table 2, PSD gains 13.6% and 4.4% improvements against MPRM and Baseline in mIoU, respectively. Compared to Zhang *et al.* [30], PSD exceeds the performance of 3.6% in terms of mIoU. The qualita-

Set.	Methods	S3DIS	ScanNet	Sem3D
Fully	PointCNN ('18) [13]	65.4	45.8	-
	DGCNN ('19) [22]	56.1	-	-
	PointConv ('19) [24]	-	55.6	-
	ShellNet ('19) [31]	66.8	-	69.3
	KPConv ('19) [20]	70.6	68.4	73.1
	PointGCR ('20) [14]	-	60.8	69.5
	RandLA-Net ('20) [5]	70.0	57.8*	77.4
	SPH3D ('20) [10]	68.9	61.0	-
	PointASNL ('20) [27]	68.7	63.0	-
	Point2Node ('20) [4]	70.0	-	-
sub.	MPRM ('20) [23]	-	41.1	-
1%	Zhang *et al.* ('21) [30]	65.9	51.1	72.6
1%	Baseline	-	50.3	68.9
1%	PSD	68.0	54.7	75.8

Table 2. Quantitative results on 6-fold of S3DIS [1], ScanNet-v2 [2], and Semantic3D (reduced-8) [3] denoted as Sem3D. "*" denotes the results of the method trained by us using the official code.

Original point cloud Semantic segmentation (1%)

Figure 3. Qualitative results on ScanNet-v2.

tive results are shown in Figure 3. It is observed that PSD achieves good segmentation results.

Results on Semantic3D. From the comparison of the quantitative results on Semantic3D (reduced-8) in Table 2, it is observed that PSD gains 6.9% and 3.2% over Baseline and Zhang *et al.* [30], respectively. Moreover, PSD achieves the 75.8% mIoU, which outperforms the fully supervised ShellNet [31] and PointGCR [14] by the gain of over 6.5% and 6.3%. These results show that PSD is also effective on the outdoor dataset.

	Base.	Aug.	Self-dis.	CAM	1pt	1%	100%
#1	✓				40.7	58.6	63.0
#2	✓	✓			41.1	59.9	63.3
#3	✓		✓		46.9	62.0	63.9
#4	✓		✓	✓	**48.2**	**63.5**	**65.1**

Table 3. The effectiveness of defferent components on Area-5 of S3DIS [1].

4.3. Ablation study

In this section, we disassemble the PSD framework and analyze some important components. All experiments are performed on Area-5 of S3DIS [1] and the results are shown in Tabel 3.

Vainness of data augmentation. To verify that the improvement is caused by PSD rather than the data augmentation, we compare the baseline (**Base.**) with the augmentation achieved by point perturbation (**Aug.**). Comparing #1 and #2 in Table 3, we find that the baseline (**Base.**) achieves the similar performance to augmentation. The results indicate that the simple point perturbation as data augmentation has a negligible effect to the results.

Effectiveness of self-distillation. We only introduce the perturbed self-distillation (**Self-dis.**) for semantic segmentation. From the comparison between #1 and #3, it can be seen that **Self-dis.** achieves significant improvements. It achieves 6.2% and 3.4% gains over Baseline at the 1pt and 1% settings, respectively. For the fully supervised setting, PSD gains 0.9% mIoU but is inferior to the weakly supervised task because PSD provides the supervision for unlabeled points by the self-distillation loss, while Baseline cannot. For the fully supervised task, sufficient label information makes the performance improvements brought by self-distillation less obvious.

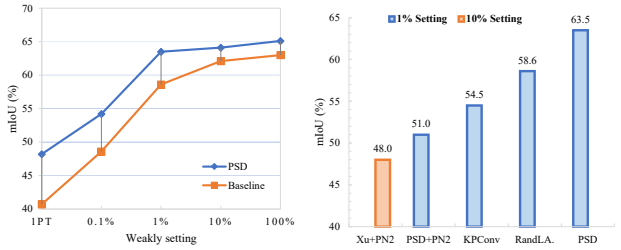
Effectiveness of context-aware module. From the comparison of #3 and #4, the context-aware module (**CAM**) gains 1.3%, 1.5%, and 1.2%, respectively. These results show that the context-aware module can further improve the performances of the weakly and fully supervised tasks.

Effectiveness of PSD. From the comparison of #1 and #4, PSD gains about 7.5%, 4.9%, and 2.1% over Baseline. This results demonstrates that PSD (**Self-dis. + CAM**) gains significant benefits from the self-distillation mechanism and context-aware module.

4.4. Analysis

Labeled points and the performance. We further discuss the relation between the number of labeled points and the segmentation performance in Figure 4 (a). With the increase of labeled points, the performance of PSD is also gradually improved, and the growth trend is gradually slowing down. These results demonstrate that sufficient labels

allow the network to learn a better representation. While for the weakly supervised task, the network requires additional supervision and accurate contextual information to improve its learning ability.



(a) (b)

Figure 4. The ablation study on different setting and backbone. (a)The relation between the number of labeled points and performance.(b) PSD is independent to backbone.. "+PN2" denotes the method use the PointNet++ [16] as backbone. KPConv and RandLA are the methods of KPConv [20] and RandLA-Net [5] at 1% settings, respectively.

Visualization of semantic correlation. We train the network by 1% labeled points and choose three scenes from Area-5 of S3DIS for visualization. The learned point affinity and the point correlation descriptor are shown in Figure 5. It demonstrates that PSD can learn an accurate affinity context. The accurate context forces the network to refine the graph topology. The practical information flow between labeled and unlabeled points will be realized, resulting in enhancing the discriminability of features.

Backbone independent. We further conduct experiments to analyze that the improvement of PSD is not attributed to the backbone. The results are shown in Figure 4 (b). When we choose PointNet++ [16] as the backbone (PSD+PN2), PSD still achieves the 51.0% mIoU at the 1% setting which is even higher than the results of Xu and Lee (48.0%) with 10% labeled points. Besides, we conduct experiments at the 1% setting for two methods: KPConv [20] and RandLA-Net [5], which achieve good performances at fully-supervised manner. It is observed that PSD still achieves the best performance. Therefore, PSD is a general framework which can be instantiated with other deep segmentation models for point clouds.

Model complexity. RandLA-Net [5] is an efficient large-scale point cloud semantic segmentation method, and PSD is built using RandLA-Net as the backbone. We list the training time of per-epoch, the network parameters, and the total test time in Table 4. Since the parameters of the two branches are shared, only the parameters of GCNs are added compared to the RandLA-Net, so that the parameters of PSD are more by 0.05M than RandLA-Net. Since the perturbed branch is only introduced in the training phase, the training time of PSD is 86s per-epoch longer than RandLA-Net. While the total test time is basically similar. Therefore, PSD is also an efficient method.

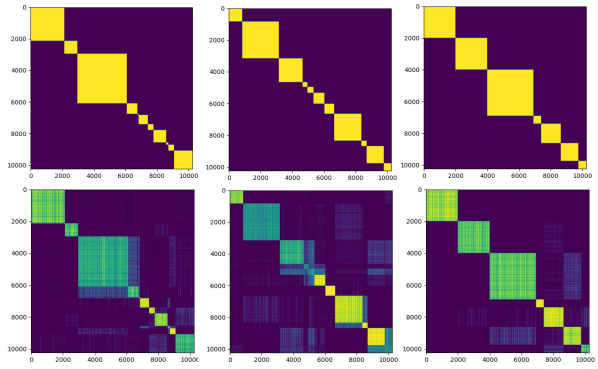


Figure 5. Visualization of the point correlation descriptor (the top row) and the predicted point affinity (the bottom row). We randomly selected 10,240 points for visualization.

Method	Training time	Network parameters	Total test time
RandLA-Net [5]	216	1.05	258
PSD (1%)	302	1.10	263

Table 4. The training time of per-epoch (in seconds), the network parameters (in millions) and total test time (in seconds) on S3DIS.

5. Conclusion

In this paper, we propose a perturbed self-distillation framework for weakly supervised large-scale point cloud semantic segmentation. Our method focuses on providing additional supervision by perturbed self-distillation to establish graph topology for implicit information propagation. Extensive experimental results demonstrate that PSD achieves significant gains compared with the state-of-the-art methods. Moreover, the effectiveness of the introduced two key components (*i.e.,* the perturbed self-distillation and context-aware module) is verified by ablation studies. The results further demonstrate that additional supervision and graph topology learning are important to improve weakly supervised semantic segmentation for large-scale point clouds.

6. Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61876161, Grant 61772524; the National Key Research and Development Program of China No. 2020AAA0108301; Natural Science Foundation of Shanghai No.20ZR1417700; Shanghai Science and Technology Commission No.21511100700; CAAI-Huawei Mind-Spore Open Fund; the Fundamental Research Funds for the Central Universities.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016.

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[3] Timo Hackel, N. Savinov, L. Ladicky, Jan D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 91–98, 2017.

[4] Wenkai Han, Chenglu Wen, Cheng Wang, Xin Li, and Qing Li. Point2node: Correlation learning of dynamic-node for point cloud feature modeling. In *CVPR*, pages 10925–10932, 2020.

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

[6] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *CVPR*, pages 984–993, 2018.

[7] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2016.

[8] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, pages 4558–4567, 2018.

[9] Huan Lei, Naveed Akhtar, and Ajmal Mian. Octree guided cnn with spherical kernels for 3d point clouds. In *CVPR*, pages 9631–9640, 2019.

[10] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[11] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *ICCV*, pages 9267–9276, 2019.

[12] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *CVPR*, pages 9397–9406, 2018.

[13] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, pages 820–830, 2018.

[14] Yanni Ma, Yulan Guo, Hao Liu, Yinjie Lei, and Gongjian Wen. Global context reasoning for semantic segmentation of 3d point clouds. In *CVPR*, pages 2931–2940, 2020.

[15] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[16] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017.

[17] Dario Rethage, Johanna Wald, Jurgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *ECCV*, pages 596–611, 2018.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[19] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, pages 1195–1204, 2017.

[20] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6411–6420, 2019.

[21] Haiyan Wang, Xuejian Rong, Liang Yang, Jinglun Feng, Jizhong Xiao, and Yingli Tian. Weakly supervised semantic segmentation in 3d graph-structured point clouds of wild scenes. *arXiv preprint arXiv:2004.12498*, 2020.

[22] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[23] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *CVPR*, pages 4384–4393, 2020.

[24] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9621–9630, 2019.

[25] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *AAAI*, pages 5565–5572, 2019.

[26] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, pages 13706–13715, 2020.

[27] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *CVPR*, pages 5589–5598, 2020.

[28] 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 *CVPR*, pages 3323–3332, 2019.

[29] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *CVPR*, pages 12416–12425, 2020.

[30] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *AAAI*, 2021.

[31] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *ICCV*, pages 1607–1616, 2019.