Computers & Graphics (2021)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

A Weakly Supervised Framework for Real-world Point Cloud Classification

An Deng[a], Peng Zhang[a], Zhuheng Lu[a], Weiqing Li[b], Zhiyong Su[a,*]

[a]School of Automation, Nanjing University of Science and Technology, Nanjing, 210094, China
[b]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China

ARTICLE INFO

Article history:
Received August 2, 2021

Keywords: Weakly supervised learning, Real-world point cloud, Point cloud classification, Point cloud segmentation

ABSTRACT

Real-world point cloud objects pose great challenges in point cloud classification as objects acquired by scanning devices from real-world scans are often cluttered with background, and are partial due to occlusions as well as reconstruction errors. In the literature, few works tackle the problem of real-world point cloud classification while existing methods require fully point-level annotated training samples. However, large-scale dense point-level foreground-background labeling for real-world point clouds is a labor-intensive and time-consuming job. In this paper, we propose a novel weakly supervised classification framework, named WSC-Net, for real-world point cloud objects. Leveraging two auxiliary modules, called semi-supervised point-level pseudo labels generation and noise-robust multi-task loss, the framework can integrate well with existing supervised point cloud classification network. A relational graph convolutional network on the local and non-local graph (PointRGCN) is first proposed to predict point-level foreground-background pseudo labels for each object with sparse ground-truth point-level foreground-background labels in training datasets. Then, a weakly supervised classification network, which combines with an auxiliary foreground-background segmentation branch, is employed to classify real-world point clouds. To cope with noise-containing point-level foreground-background labels generated above, a noise-robust multi-task loss is proposed to train the network accurately. Experimental results show that the performance of the proposed framework which trained with even only 1% point-level labels is comparable with many popular or state-of-the-art fully supervised methods. The source code can be found at http://zhiyongsu.github.io.

1. Introduction

Point cloud object classification, which is a classical and critical problem in computer graphics and computer vision fields, aims to identify the categories of different point cloud objects [1, 2]. The rapid development of scanning devices have witnessed the widely application of point clouds in the fields of robotics, autonomous vehicles, augmented reality, urban planning, industrial manufacturing applications, etc [3, 4, 5]. Many works in the literature have made great progress in the synthetic 3D point cloud classification task [1, 6, 7, 8, 9, 10]. The overall accuracy of the state-of-the-art methods on ModelNet40, the most popular synthetic point cloud dataset in point cloud object classification, has reached more than 93% in 2020 [5], and the trend of bringing the accuracy towards perfection is still ongoing.

However, recent studies show that the classification models trained on synthetic data often do not generalize well to real-world point cloud objects, and vice versa [11, 12]. Synthetic datasets are usually developed with the assumption that objects are complete, clean, and especially free from any background

*Corresponding author.
e-mail: su@njust.edu.cn (Zhiyong Su)

noise. Unfortunately, real-world point cloud objects, which are usually obtained in real-world settings through LiDAR sensors, or RGBD scanners, may suffer from background points (surroundings), noise, and holes. These real-world objects will introduce some confusing information, which increases the difficulty of classifying real-world point cloud. Consequently, applying existing point cloud classification methods to real-world objects may not achieve the same good results as synthetic data. Therefore, how to handle background effectively when they appear together with objects due to clutter in the real-world scenes is still a very challenging task [11].

Up to now, only a few pieces of works target the real-world point cloud classification problem. They attempt to deal with this challenging task from the perspectives of transfer learning [12], learning transformation invariant representation [13, 14], and multi-task learning [11]. Considering the limited amount of annotated real-world point cloud data, transfer learning based methods attempt to employ extra synthetic data to enrich standard feature representations [12]. Since the real-world point clouds are not well aligned, several works try to learn translation and rotation invariant point cloud features by the attention mechanism to improve the robustness against translation and rotation [13, 14]. However, none of these works have considered the problem of background points which is the most challenging task of real-world point cloud classification. Multi-task learning based method benefits the real-world point cloud classification network through an auxiliary fully-supervised segmentation task, which is employed to distinguish the foreground and background points [11]. However, it requires accurate dense point-level foreground-background labels(annotations). Despite recent developments of modern annotation toolkits [15, 16], exhaustive labeling is still a quite labor intensive and time-consuming job for ever-growing new datasets.

In this paper, we propose a novel weakly supervised classification framework, called WSC-Net, for classifying real-world point clouds. The concept of weak supervision in this paper contains incomplete supervision and inaccurate supervision [17]. Specifically, the framework is designed to take advantage of and integrate well with existing supervised point cloud classification network through introducing two auxiliary modules : semi-supervised point-level pseudo labels generation and noise-robust multi-task loss. The former aims to generate point-level foreground-background pseudo labels for each object in training datasets with sparse ground-truth point-level foreground-background labels, as illustrated in Fig.1. Undoubtedly, noisy labels will be inevitably introduced during this stage. The latter strives to fade away their negative effects to train the classification network efficiently and accurately. Compared with existing real-world point cloud classification methods, our method can yield competitive performance without the need of tedious and time-consuming labeling processes for preparing training data. Therefore, it is much more practical than existing fully-supervised approaches.

In summary, the main contributions of this paper are as follows:

- A novel weakly supervised framework for real-world point cloud object classification is proposed. The framework can make full use of existing supervised point cloud classification network by incorporating semi-supervised pseudo labels generation and noise-robust multi-task loss.

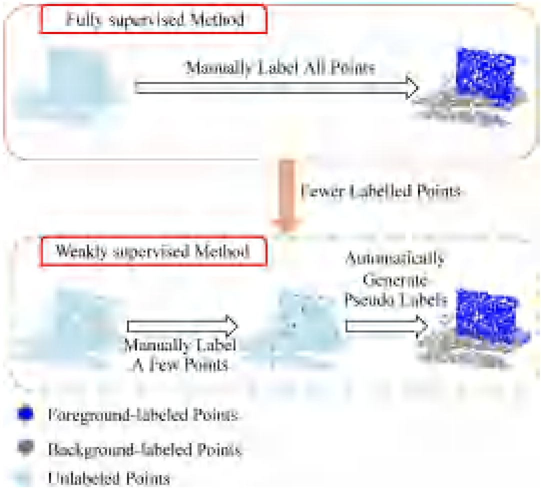

Fully supervised Method
Manually Label All Points
Fewer Labelled Points
Weakly supervised Method
Manually Label A Few Points
Automatically Generate Pseudo Labels
Foreground-labeled Points
Background-labeled Points
Unlabeled Points

Fig. 1. Illustration of the real-world point cloud contaminated by background points, and the weak supervision concept in this work. Our weakly supervised approach assists real-world point cloud classification with fewer foreground-background labeled points.

- A local and non-local graph based relational graph convolutional network (PointRGCN) is proposed to generate point-level foreground-background pseudo labels for each object in training datasets in a semi-supervised manner. By applying the PointRGCN, each point can aggregate more discriminative features from multi-type of neighbors, resulting in producing more accurate point-level pseudo labels.

- A noise-robust multi-task loss that combines classification and segmentation losses is proposed to train the classification network accurately with noisy-containing point-level foreground-background labels.

- We demonstrate that our proposed WSC-Net framework produces comparable results with the state-of-the-art fully-supervised approaches with even only 1% point-level labels.

The remaining of this paper is organized as follows: In Section 2, we compare the difference between synthetic and real-world point cloud classification work and then discuss the work related to weakly supervised learning on point cloud and noisy label learning. Section 3 gives an overview of the proposed weakly supervised real-world point cloud classification framework. Problem Formulation is discussed in Section 4. We introduce the proposed PointRGCN and noise-robust multi-task loss in detail in Section 5 and 6, respectively. After that, we present the experimental results of the proposed weakly supervised real-world point cloud classification method in Section 7. Finally, conclusions and suggestions for future research are provided in Section 8.

2. Related Work

2.1. *Synthetic Point Cloud Classification*

Early attempts at point cloud classification generally focused on the ideal synthetic point cloud data. Overall, synthetic point cloud classification methods can be subdivided into projection-based methods, voxel-based methods, graph-based methods, and point-convolution-based methods. Projection-based methods need to project the original point cloud into 2D images and use 2D CNNs to process them [18, 19, 20]. These methods are also called multi-view based methods. Since multi-view based methods need to render batches of 2D images, it will lose the intrinsic geometric features of the point cloud. Voxel-based methods usually voxelize point clouds into 3D grids, then feed them to CNNs [21, 22]. However, the 3D CNNs are very computationally expensive, hence the resolution of point clouds is highly limited. Alternatively, point-based methods can directly handle 3D point clouds. According to the different ways of point feature learning, these methods can be divided into MLPs-based methods, graph-based methods, and point convolution based methods. MLPs-based methods represented by PointNet [1] and its variants [6, 23, 7] learn 3D point cloud feature through multi-layer perceptrons (mlps), capture local geometric features by aggregating neighboring information, and use symmetric functions to aggregate the features of all points to form a global shape descriptor. Graph-based methods [24, 25, 26, 27, 28, 29] consider the points in the point cloud as nodes in a graph, and construct links between points as directed edges. Therefore, the graph neural network defined in the spectral domain or the spatial domain is introduced to process point cloud objects. Recently, Point convolution operators [8, 10, 30, 9, 31, 32] are proposed to apply convolution operations on point clouds directly.

Nevertheless, the aforementioned point cloud classification methods only focus on synthetic point cloud data and do not consider real-world point cloud data contaminated by background points. Due to the characteristics of real-world point clouds, the existing work that tends to be perfect on ideal synthetic point cloud data cannot work well on real-world data.

2.2. *Real-world Point Cloud Classification*

To the best of our knowledge, currently, there are only a handful of works that consider the classification of real-world object point cloud. Uy et al. [11] propose the background-aware (BGA) model to handle the occurrence of background points in point clouds obtained from real scans, but this method requires dense point-level foreground-background labels. A method that joints supervised and self-supervised learning is proposed in [12]. It enriches the point cloud features by jointly learning a supervised main classification task and a self-supervised 3D puzzle which is to reassemble the split point cloud. Since it does not directly address the impact of background points on real-world point cloud classification tasks, the performance is still unsatisfying. Zhao et al. [14] propose combining local geometry with global topology toward achieving rotation-invariant representations for the real-world point cloud. Fuchs et al. [13] present an attention-based neural architecture that is robust of rotations and translations of the point cloud. It updates the point features using SE(3)-Transformer, an equivariant attention mechanism. These two works have shown great potential in improving the robustness to the geometric transformation, but none of them tries to handle point clouds contaminated by the background.

In short, existing real-world point cloud data classification methods either require complete foreground-background annotation point clouds as training data, or ignore the interference caused by background points to point cloud classification. Our method attempts to combine weakly supervised learning with real-world point cloud classification tasks. We try to handle real point clouds contaminated by background points by taking sparse point-level annotations as supervision.

2.3. *Weakly Supervised Learning on Point Cloud*

In this paper, we focus on point cloud feature learning with less supervision. Therefore, we will separately discuss unsupervised, self-supervised, and weakly supervised learning on the point cloud. Unsupervised work does not require labeled data, but most of them focus on low-level visual tasks, such as point clustering, distinctive region detection, etc. Li et al. [33] presents an approach to learn and detect distinctive regions on 3D shapes in an unsupervised manner. Self-supervised work mainly focuses on point-wise feature learning and cannot be directly used for downstream point cloud classification and segmentation tasks. MortonNet [34] learns point-wise features by leveraging space-filling curves in a self-supervised manner. Sauder et al. [35] propose a self-supervised learning task to learn point cloud representations by training a model that reassembles randomly split point clouds. Recently, several works have tried to utilize less supervision to achieve point cloud semantic segmentation. A weakly supervised point cloud segmentation method is proposed in [36] by introducing additional losses to regularize the model. However, it adds a smooth branch to the original segmentation network, which will reduce the efficiency of the inference stage. Wei et al. [37] propose a weakly supervised point cloud segmentation method that only uses scene-level weak labels and subcloud weak labels. Yet this work only focuses on large scenes with many different types of objects, it dose not extend to point cloud part segmentation tasks.

It is hard to directly apply these methods to process the real-world point cloud. Meanwhile, these methods are difficult to directly and efficiently integrate into the existing point cloud classification approaches. We propose a weakly supervised point cloud segmentation method based on the semi-supervised pseudo labels generation and noisy label learning to assist the real point cloud classification network.

2.4. *Robust Learning with Noisy Labels*

In practical applications, it is unrealistic to always obtain completely clean labeled data. To this end, many approaches have been proposed to learn with noisy labels, such as correcting noise labels, using adaptive training strategies, modifying loss functions, using noise-robust loss function, etc. There are

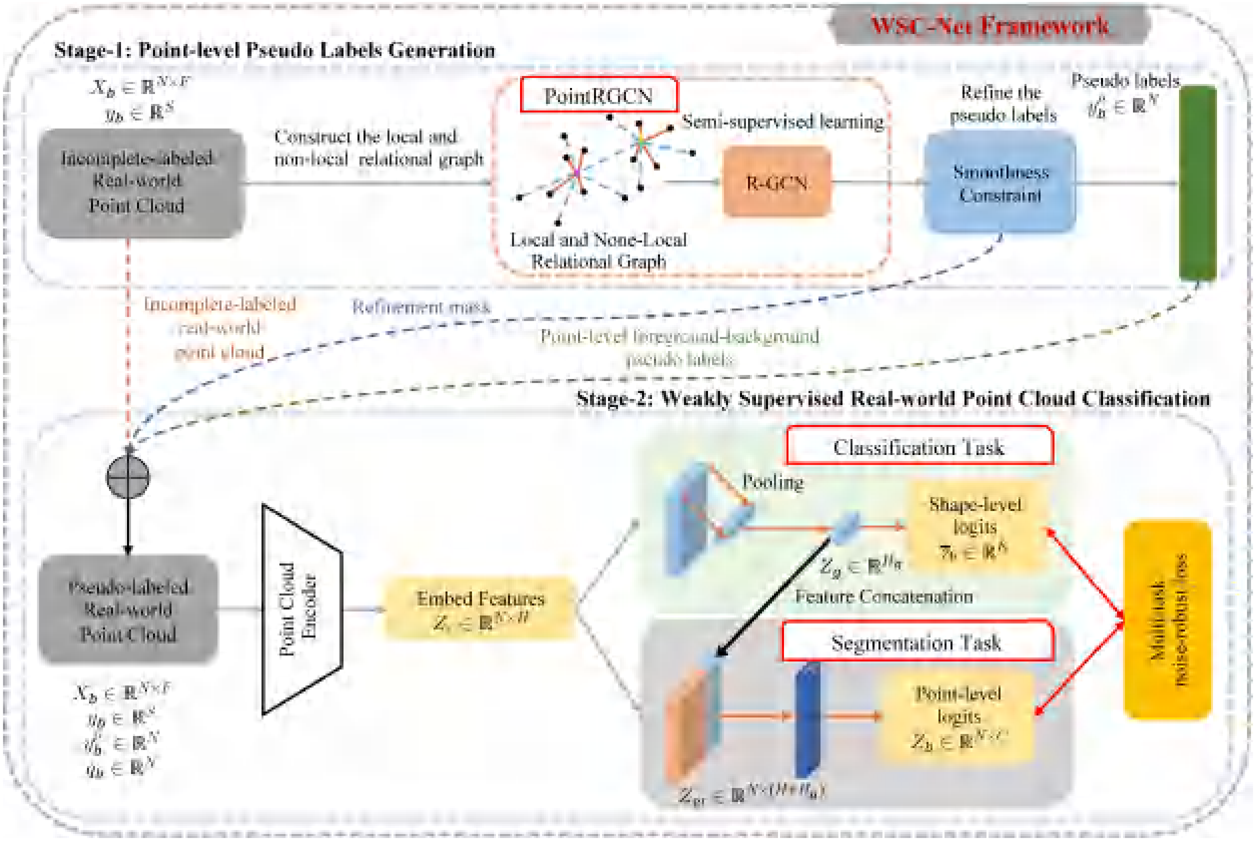

WSC-Net Framework
Stage-1: Point-level Pseudo Labels Generation
$X_b \in \mathbb{R}^{N \times F}$
$y_b \in \mathbb{R}^{S}$
Incomplete-labeled Real-world Point Cloud
Construct the local and non-local relational graph
PointRGCN
Semi-supervised learning
R-GCN
Local and None-Local Relational Graph
Refine the pseudo labels
Smoothness Constraint
Pseudo labels
$y_b^{\prime} \in \mathbb{R}^{N}$
Incomplete-labeled real-world point cloud
Refinement mask
Point-level foreground-background pseudo labels
Stage-2: Weakly Supervised Real-world Point Cloud Classification
Pseudo-labeled Real-world Point Cloud
$X_b \in \mathbb{R}^{N \times F}$
$y_b \in \mathbb{R}^{S}$
$y_b^{\prime} \in \mathbb{R}^{N}$
$y_b \in \mathbb{R}^{N}$
Point Cloud Encoder
Embed Features
$Z_i \in \mathbb{R}^{N \times H}$
Classification Task
Pooling
Shape-level logits
$Z_b \in \mathbb{R}^{K}$
$Z_g \in \mathbb{R}^{H_g}$
Feature Concatenation
Segmentation Task
Point-level logits
$Z_b \in \mathbb{R}^{N \times C}$
$Z_{gr} \in \mathbb{R}^{N \times (H+H_g)}$
Multi-task Noise-robust loss

Fig. 2. The proposed weakly supervised real-world point cloud classification framework.

several attempts using conditional random field [38], neural network [39], knowledge graph [40], and other methods to correct the wrong labels. However, those approaches require additional clean label data. Another scenario is to design adaptive training strategies that are more robust to the noisy labels [41, 42, 43]. Some approaches enhance the robustness of the model to noisy labels by modifying the loss function. Han et al. [44] estimate a noise transformation matrix that defines the probability of mislabeled classes with others classes [45]. It has been proved that the Mean Absolute Error (MAE) is robust to noisy labels, but the commonly used Cross Entropy (CE) loss is not [46]. However, the robustness of MAE will increase the difficulty of training. The Generalized Cross Entropy (GCE) loss proposed in [47] can be seen as a generalization of MAE and CE. Wang et al. [48] propose the Symmetric Cross Entropy (SCE) which combines a Reverse Cross Entropy (RCE) with the CE loss, and it strikes a balance between sufficient learning and robustness to noisy labels.

For learning from noisy pseudo labels, we propose a smooth constraint to filter suspicious pseudo labels. Then we adopt point cloud segmentation loss to a noise-robust loss. Lastly, we utilize classification loss and noise-robust segmentation loss to construct a noise-robust multi-task loss. Thus, we take advantages of both noise-robust learning and real-world point cloud geometric characters.

3. Overview of the framework

The proposed WSC-Net framework for real-world point cloud classification consists of two stages, as illustrated in Fig.2. In the first stage, to produce extra supervision from limited labeled points, we propose a novel semi-supervised PointRGCN to generate pseudo foreground-background labels for each incompletely labeled real-world point cloud object. A *k*-NN voting based smoothness constraint module is introduced to refine the pseudo labels, since the generated pseudo labels contain some misclassifications. In the second stage, a weakly supervised real-world point cloud classification network is employed to classify real-world point cloud. To accurately train the final classification model, a novel multi-task combined noise-robust loss constrained on shape-level labels, point-level sparse labels, and noisy pseudo labels is introduced. The filtered points are only used for feature extraction, and the loss of these points is ignored.

4. Problem Formulation

We consider a point cloud dataset with B samples and K classes as $\{\mathbf{X}_b\}_{b=1,2,3,...,B}$. Each sample $X_b \in \mathsf{R}^{N \times F}$ consists of N 3D points with its xyz coordinates and other additional attributes, e.g., surface normal, RGB values. Each sample X_b is accompanied with a shape-level class label $V_b \in \{1, ... K\}$ which is easy to obtain, and it is further accompanied with ground-truth point-level foreground-background la-

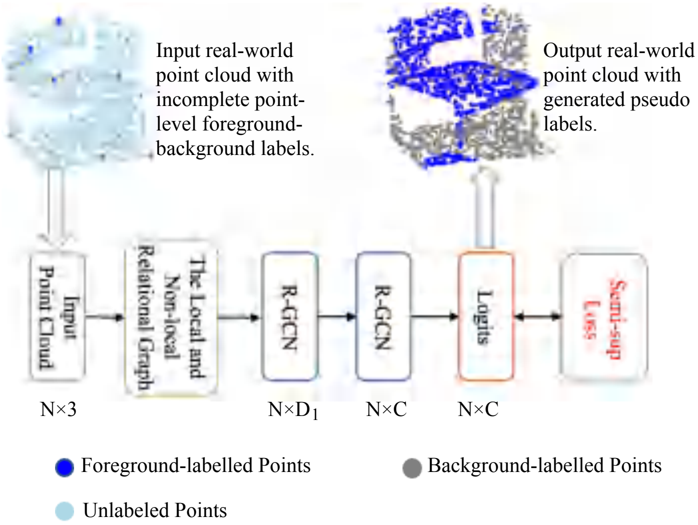

Input real-world point cloud with incomplete point-level foreground-background labels.
Output real-world point cloud with generated pseudo labels.
Input Point Cloud
The Local and Non-local Relational Graph
R-GCN
R-GCN
Logits
Semi-sup Loss
N×3
N×D1
N×C
N×C
● Foreground-labelled Points
● Background-labelled Points
● Unlabeled Points

Fig. 3. An illustration of the proposed PointRGCN for point-level pseudo labels generation with a per-point loss function.

bel $y_b \in \{0,1\}^S$, where 0 and 1 represent the background and foreground point respectively, S is the number of points with ground-truth point-level labels, in this paper, $S \ll N$. We denote the one-hot encoded shape-level label and point-level labels as $\hat{V}_b$ and $\hat{y}_b$, respectively. To get an incompletely labeled real-world point cloud dataset, we assume that only S points of each training sample are labeled. Therefore, we define a binary mask $M \in \{0,1\}^{B\times N}$. For each point x_i in sample X_b, $m_{bi} = 0$ means that x_i is unlabeled, otherwise it has a ground-truth label. A semi-supervised relational graph convolutional neural network $h(\mathbf{X};\Theta_1)$, parameterized by Θ_1, is designed to generate pseudo labels for each sample X_b. A point cloud encoder network $f(\mathbf{X};\Theta_2)$, e.g., DGCNN [7], PointNet++ [6], parameterized by Θ_2, is employed to obtain the embedded point cloud features $Z_e \in \mathrm{R}^{N\times H}$.

5. Point-level Pseudo Labels Generation

In this section, to produce extra supervision for the classification model training, we firstly propose a PointRGCN to generate point-level foreground-background pseudo labels for the incompletely labeled real-world point cloud, as shown in Fig. 3. The proposed PointRGCN consists of two parts: the local and non-local relational graph construction and a semi-supervised relational graph convolutional network (R-GCN). A smoothness constraint module is introduced to filter suspicious pseudo labeled points.

5.1. The Local and Non-Local Relational Graph Construction

To extract more discriminative point cloud features, we construct a local and non-local relational graph to aggregate local and non-local information of each point, as shown in Fig. 4. Currently, the common methods used to construct the graph for point cloud, such as searching the k-nearest neighbors in Euclidean space [1, 7] (k-NN graph) or searching neighbors in a fixed metric radius sphere [6, 24] (radius graph), can only aggregate point features in a local way. As a result, the extracted features of points on the local graph tend to be smoothed. Therefore, to avoid over-smoothing features and generate pseudo labels with higher quality, for each point, we choose to not only aggregate its k-nearest neighbors in Euclidean space, but also aggregate the points which have similar local geometric features but located distantly in Euclidean space [2].

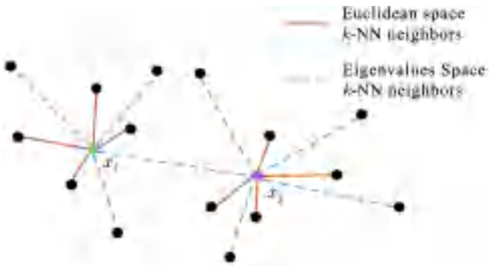

Euclidean space k-NN neighbors
Eigenvalues Space k-NN neighbors
x_i
x_j

Fig. 4. An illustration of the local and non-local relational graph, different colored lines indicate different types of neighborhoods.

Firstly, we use the k-NN algorithm to get the k_1-nearest neighbors of each point x_i in Euclidean space, written as $\mathcal{N}_i^{local}$. Then, following [2], we get the k_2 non-local neighbors of each point x_i in eigenvalues space, written as $\mathcal{N}_i^{non-local}$. The k_1-nearest neighbors of point x_i is $\{x_{i1}, x_{i2}, ..., x_{ik_1}, x_{ik_1} \in \mathcal{N}_i^{local}\}$. Let $U = \{x_{i1} - x_i, x_{i2} - x_i, ...x_{ik_1} - x_i\}$, then define $C = U^T \times U$. We have the decomposition $C = R\Lambda R^T$, where R is the rotation matrix, and Λ is a diagonal and positive definite matrix, known as eigenvectors and eigenvalues matrixes respectively [2]. Get the eigenvalues of each point x_i which can be represented as $\lambda_i \in \mathrm{R}^3$, and ordered as $\lambda_i^1 \geq \lambda_i^2 \geq \lambda_i^3$, to form the eigenvalues space. We use L^2 distance to calculate the distance in eigenvalues space between different points. Then, we choose the k_2 nearest neighbors for each point x_i according to:

$$Distance(x_i, x_j) = \|\lambda_i - \lambda_j\|. \tag{1}$$

Finally, by properly arranging these two types of neighbors, each point can capture richer local and non-local information. We unite the local neighbors $\mathcal{N}_i^{local}(1 \leq i \leq N)$ and non-local neighbors $\mathcal{N}_i^{non-local}(1 \leq i \leq N)$ to form the local and non-local relational graph, represented as $\mathcal{N}_i^r(1 \leq i \leq N, r \in \{local, non-local\})$.

We define the adjacency matrix of the local and non-local relational graph as:

$$A = \begin{bmatrix} c_{00} & \cdots & c_{0N} \\ & \vdots & \\ c_{N0} & \cdots & c_{NN} \end{bmatrix}, \tag{2}$$

where c_{ij} indicates whether there is a relation between x_i and x_j. $c_{ij} = 1$ if x_j is the neighbor of x_i, otherwise $c_{ij} = 0$. According to the different types of neighborhood relations between x_i and x_j, we can build the edge type matrix E as:

$$E = \begin{bmatrix} e_{00} & \cdots & e_{0N} \\ & \vdots & \\ e_{N0} & \cdots & e_{NN} \end{bmatrix}, \tag{3}$$

Algorithm 1: PointRGCN, Point-level Pseudo Labels Generation

```
Input: Point Cloud $\{X_b \in \mathsf{R}^{N\times F}\}$, Labels $\{y_b \in \mathsf{Z}^S\}$
Output: Pseudo Labels Predictions $\{y_b^p \in \mathsf{Z}^N\}$
/* The Local and Non-local Relational Graph
   Construction:                                          */
Search $k_1$-nearest neighbors in Euclidean space to get local
  neighbors;
Search $k_2$-nearest neighbors in eigenvalue space to get non-local
  neighbors;
Construct the local and non-local relational graph according to
  Eq. (2) and Eq. (3);
/* Smei-Supervised R-GCN                                  */
for $Epoch \leftarrow 1$ to $Epochs$ do
    Train the R-GCN one epoch: $\Theta_1 = \Theta_1 - \alpha\nabla\mathcal{L}_{semi}|_{\{X_b\},\{y_b\}}$;
end
/* Obtain Predictions:                                    */
Forward pass $Z_h = h(X_b; \Theta_1)$
Obtain pseudo labels predictions $y_b^p$ via $\mathrm{argmax}_i\, z_{hi}$ ;
```

where e_{ij} denotes the relational type of x_i and x_j. We have $e_{ij} \in \mathcal{E}$, and the types of all relations in the local and non-local relational graph form the relation types set $\mathcal{E} = \{local, non-local\}$.

5.2. Semi-supervised Relational Graph Convolutional Networks

In order to extract richer information from multiple relation types of neighbors for each point, we introduce a relational graph convolutional network (R-GCN) [49] to generate pseudo foreground-background labels in a semi-supervised manner. R-GCN uses multiple groups of weights to learn feature transformations between different relation types. The proposed PointRGCN consists of two R-GCN layers (defined in Eq.(4)), and the output of the previous layer is the input to the next layer [49]. In the first layer, it takes the local and non-local relational graph as input, then extracts high-level point features $Z_h^1 \in \mathsf{R}^{N\times D}$. For the second layer, the embedded features Z_h^1, after passing a ReLU activation, is mapped to prediction scores $Z_h \in \mathsf{R}^{N\times 2}$. The forward propagation formula of the R-GCN layer can be written as:

$$Z_{hi}^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{E}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}Z_{hj}^{(l)} + W_0^{(l)}Z_{hi}^{(l)}\right), \quad (4)$$

where $Z_{hi}^{(l)}$ is the hidden state of point x_i in the l-th layer, $j \in \mathcal{N}_i^r$ denotes the set of neighbors of point x_i under the relation type r, $c_{i,r}$ is a problem-specific normalization constant which is set to $|\mathcal{N}_i^r|$ in this paper, and W is a learnable weight matrix.

The optimize objective Eq.(5) is penalized only on those ground-truth labeled points, while ignoring unlabeled points. We minimize the per-point softmax cross-entropy loss only on labeled points:

$$\mathcal{L}_{semi} = -\sum_i m_{bi}\sum_k \hat{y}_{bik}\log\frac{\exp(z_{hik})}{\sum_k \exp(z_{hik})}, \quad (5)$$

where m_{bi} indicates whether point x_i is labeled or not, which means the semi-supervised R-GCN is constrained with only a few labeled points $\hat{y}_b \in \{0,1\}^{\{S\times 2\}}$, z_h is the logits on foreground and background. Through multiple epochs of training, each point can obtain more discriminative fine-grained features from the local and non-local relational graph, and the unlabeled points can get the predicted pseudo labels.

The proposed pseudo labels generation algorithm is shown in Algo.1.

Algorithm 2: Smoothness Constraint

```
Input: Point Cloud $\{X_b \in \mathsf{R}^{N\times F}\}$, Pseudo Labels $\{y_b^p \in \mathsf{Z}^N\}$
Output: Refinement Mask $\{Q_b \in \mathsf{R}^N\}$;
for $x_i : X_b$ do
    /* Obtain Majority Predictions                          */
    Search $k$-nearest-neighbors in euclidean space for $x_i$;
    Find the most frequent label in the collection of $x_i$'s $k$-NN
      neighbors' pseudo labels $\mathbf{y}_{\mathcal{N}_i}$, written as $y_{bi'}$(majority
      prediction).
    /* Construct Refinement Mask Based on Majority
       Prediction                                           */
    if $y_{bi'} = y_{bi}^p$ then
        $q_{bi} = True$;
    else
        $q_{bi} = False$;
    end
end
```

5.3. Smoothness Constraint

A k-NN voting based smoothness constraint module is introduced to filter out the suspicious pseudo labels, and further refine the generated pseudo labels. Although the PointRGCN can generate high-quality predicted labels for the majority of unlabeled points, the prediction is not perfect. To alleviate the negative impact of wrong labeled points, we introduce a k-NN voting based method to filter out suspicious points. We count the number of different types of labels in the 1-hop k-NN neighborhood of each point x_i. If the pseudo label of x_i is inconsistent with the mode of its k-NN neighbors' pseudo labels, it is considered as an invalid labeled point. A binary refinement mask $\mathbf{q} \in \{0,1\}^{B\times N}$ can be obtained through the k-NN voting mechanism to indicate whether a point is valid or not. For each point x_i, it is invalid if $q_{bi} = 0$, otherwise it's valid. The k-NN voting based smoothness constraint algorithm is summarized in Algo.2.

6. Weakly Supervised Real-world Point Cloud Classification Network

In this section, we firstly present a weakly supervised real-world point cloud classification network architecture which is employed to classify real-world point cloud, with the assistance of an auxiliary weakly supervised segmentation task. Then, we present the noise-robust multi-task loss to train the classification model accurately on noisy pseudo labels.

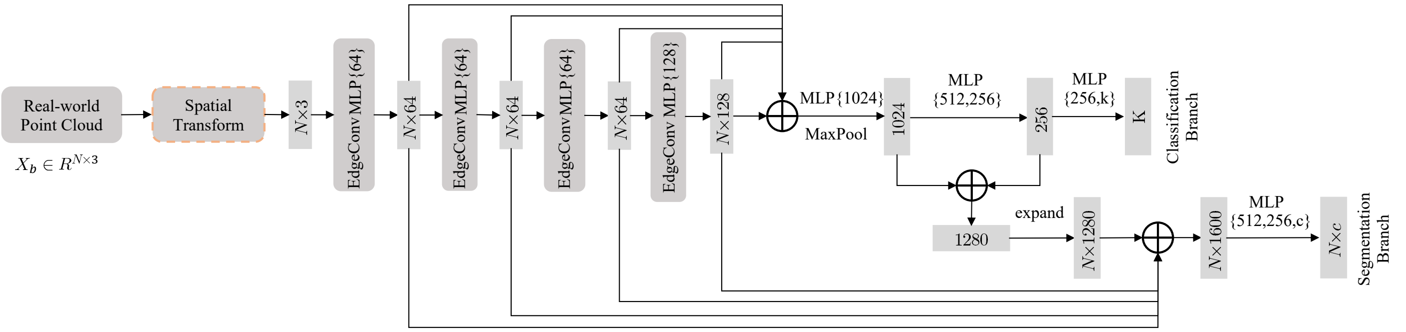


Fig. 5. Weakly Supervised Real-world Point Cloud Classification Network.

6.1. Network Architecture

The weakly supervised real-world point cloud classification network architecture, as depicted in Fig. 5, consists of a classification branch that trained on shape-level labels and a weakly supervised foreground-background segmentation branch that trained on few point-level ground-truth labels and generated pseudo labels, inspired by the fully supervised background-aware(BAG) classification network [11].

We choose DGCNN [7] as the backbone network to extract point cloud features, due to its impressive performance in both point cloud classification tasks and segmentation tasks. In the classification branch, point cloud features Z_e are aggregated into a 1-D global feature, i.e. $Z_g = \max_i z_{ei}$, to produce object classification scores $\bar{z}_b \in \mathrm{R}^K$ through several fully connected layers. As well as the segmentation branch, the expanded global feature $Z_{g'} \in \mathrm{R}^{N\times H_g}$ and point-level features Z_e are concatenated as $Z_{e'}$ to obtain point-level segmentation scores $Z_b \in \mathrm{R}^{N\times C}$.

The classification branch is constrained on the shape-level labels, and the segmentation branch is constrained on a small number of ground-truth labels and generated pseudo labels. We train both classification and foreground-background segmentation branches jointly with noisy label learning.

6.2. Noise-robust Multi-task Loss

To accurately train the weakly supervised real-world point cloud classification network, we propose a noise-robust multi-task loss that combines both classification loss and noise-robust segmentation loss. The combined classification-segmentation loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{seg}, \tag{6}$$

where λ is used to balance the classification task and segmentation task.

For the classification branch, we firstly use a channel-wise symmetric aggregation operation on the embedded feature, i.e. $z_b = max_i(z_{bi})$. Then we use several fully connected layers to obtain $\tilde{z}_b$, the logits on each object category. We apply a softmax Cross Entropy (CE) loss for the classification branch. Consequently, the classification loss can be defined as:

$$\mathcal{L}_{cls} = -\sum_k \hat{V}_{bk} \log \frac{\exp(\bar{z}_{bk})}{\sum_k \exp(\bar{z}_{bk})}, \tag{7}$$

where $\hat{V}_b$ is the one-hot encoded shape-level label.

For the segmentation branch, since the refined pseudo labels still contain a small amount of misclassification, we propose a segmentation loss $\mathcal{L}_{seg}$ which is tolerant to noisy pseudo labels. The proposed loss $\mathcal{L}_{seg}$ contains two terms: the segmentation loss on the weak labels, and the segmentation loss on the pseudo labels. Hence the segmentation loss $\mathcal{L}_{seg}$ can be written as:

$$\mathcal{L}_{seg} = \mathcal{L}_{weak} + \mathcal{L}_{pseudo}. \tag{8}$$

$\mathcal{L}_{weak}$ that penalizes weak labels can be specifically written as:

$$\mathcal{L}_{weak} = -\sum_i m_{bi} \sum_k \hat{y}_{bik} \log \frac{\exp(z_{bik})}{\sum_k \exp(z_{bik})}, \tag{9}$$

where z_b is the point-level logits on each segmentation category, m_{bi} indicates whether point x_i has a ground-truth label, and $\hat{y}_b$ is the one-hot encoded point-level labels.

Due to the corruption of pseudo labels, it's difficult to train an accurate real-world point cloud segmentation branch only with pseudo labels. Meanwhile, an inaccurate segmentation branch will not help the real-world point cloud classification, and will even degrade the classification performance. It has been demonstrated that the CE loss commonly used in point cloud segmentation task is not robust against noisy labels [46, 45]. Therefore, we introduce a noise-robust Symmetric Cross Entropy (SCE) [48] loss function to accurately train the segmentation branch with pseudo labels. The SCE loss consists of a Reverse Cross Entropy (RCE) term which is noise tolerant under symmetric or uniform label noise if noise rate $\eta < 1 - \frac{1}{K}$ and a CE term which is not, but useful for achieving good convergence. The SCE is formally written as:

$$\begin{aligned}
\mathcal{L}_{pseudo} &= \mathcal{L}_{ce} + \mathcal{L}_{rec} \\
&= -\sum_i (\neg m_{bi}) \wedge (q_{bi}) \\
&\quad \{\alpha_p \sum_k \hat{y}^p_{bik} \log \frac{\exp(z_{bik})}{\sum_k \exp(z_{bik})} + \\
&\quad \beta_p \sum_k z_{bik} \log \frac{\exp(\hat{y}^p_{bik})}{\sum_k \exp(\hat{y}^p_{bik})}\}
\end{aligned} \tag{10}$$

where ¬ means logical negation, ∧ means logical and. q_{bi} means whether point x_i is valid, that is to say, $\mathcal{L}_{pseudo}$ only penalizes the valid pseudo labeled points. $\hat{y}^p_b$ is the one-hot encoded pseudo labels. α_p and β_p are hyperparameters, with α_p

Algorithm 3: Weakly Supervised Real-world Point Cloud Classification via Pseudo Labels Generation and Noise-robust Multi-task Loss

```
Input: Training Dataset:
Point Cloud in Training Set$\{X_b \in \mathbb{R}^{N\times F}\}$ ;
A Few Point-level Labels$\{y_b \in \mathbb{Z}^S\}$
Testing Dataset:
Point Cloud in Testing Set$\{\tilde{X}_b \in \mathbb{R}^{N\times F}\}$
Output: Category Predictions$\{\tilde{V}_b \in \mathbb{Z}\}$
/* Training Stage: */
Generate point-level pseudo labels $\{y_b^p \in \mathbb{Z}^N\}$ according to Algo. 1;
Calculate refinement mask $\{Q \in \mathbb{R}^N\}$ according to Algo. 2;
for $Epoch \leftarrow 1$ to 150 do
    Train one epoch: $\Theta_2 = \Theta_2 - \alpha\nabla\mathcal{L}_{total}|_{\{X_b\},\{y_b\},\{\tilde{y}_b\},\{Q\}}$;
end
/* Inference Stage: */
Forward pass $Z_p = h(\tilde{X}_b)$;
Obtain final prediction of via $\tilde{V}_b = \arg\max_k z_{pk}$;
```

on the overfitting issue of CE while β_p for flexible exploration on the robustness of RCE [48].

The proper weights between $\mathcal{L}_{weakly}$ and $\mathcal{L}_{pseudo}$ are very critical for the final performance. Considering the number of weak labels is much less than that of pseudo labels, if we use the Eq.(8) directly, the segmentation branch will tend to fit noisy pseudo labels instead of the ground truth weak labels [50]. For the above reasons, the regularization is applied to re-weights the losses calculated on different labels to correct the impact of pseudo labels. The weighted segmentation loss function is:

$$\mathcal{L}_{seg} = \alpha\mathcal{L}_{weakly} + \beta\mathcal{L}_{pseudo}, \tag{11}$$

where α and β are parameters to balance the two terms. Then, the final segmentation loss $\mathcal{L}_{seg}$ is the weighted sum of the two terms. Since how to balance these two loss functions plays an important role on the final performance of the model [50], we assume $\beta << \alpha$ in this paper.

Finally, the classification loss and the noise-robust segmentation loss are combined as the noise-robust multi-task loss:

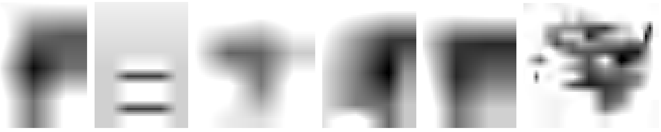
$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \lambda(\alpha\mathcal{L}_{weakly} + \beta\mathcal{L}_{pseudo}). \tag{12}$$

7. Experiment

In this section, we firstly evaluate our proposed WSC-Net on the ScanobjectNN [11] dataset. Then, we conduct detailed experiments to evaluate the importance of different modules and the compatibility with alternative backbone. We also visualize experimental results to analyze the effect of weakly supervised segmentation branch on the real-world point cloud classification network.

7.1. Implementation Details

In the point-level pseudo labels generation stage, we train the R-GCN for 64 epochs and the output in the final epoch is the predictions of pseudo foreground-background labels. To prevent over-fitting, the early stop strategy is adopted. That is to say, when the accuracy on the labeled points reaches 99% over 3 times, the training is stopped. Adam optimizer is used for efficient training. The initial learning rate is 0.0135, and we reduce the learning rate until 0.0001 using cosine annealing.



(a) Objects only. (b) Objects with background.

Fig. 6. Example objects from ScanobjectNN [11].

The final training objective is the noise-robust multi-task loss function (Eq.(12)). We use an Adam optimizer, and the initial learning rate is 0.1. The exponential decay is applied to the learning rate. The batchsize is set at 32. We train the network for 150 epochs.

The number k_1 and k_2 of nearest neighbors in the local and non-local relational graph is set to 20, respectively. In the pseudo label refining step, we set the number k of nearest neighbors to 50 for the k-NN voting. The number k of nearest neighbors in DGCNN is 20.

Our proposed weakly supervised real-world point cloud classification approach is summarized in Algo. 3.

7.2. Dataset

To the best of our konwledge, ScanobjectNN is the first and only real-world point cloud object dataset based on scanned indoor scene data with foreground-background annotations in the literature [11]. ScanObjectNN consists of two types of data, namely the object point cloud with background, and the object point cloud only, as shown in Figure 6. It is created from the state-of-the-art scene mesh datasets SceneNN and ScanNet in the way of automatically instance segmenting and manually filtering. And, it contains 15 common daily objects categories, such as tables, chairs, bookshelves, sinks, toilets, displays, etc.

Among the several variants of ScanObjectNN, we choose the most challenging one, the ScanObjectNN-PB_T50_RS, to evaluate our method. Each sample in this dataset randomly shifts the bounding box up to 50% of its size from the box centroid along each world axis. Meanwhile, rotation and scaling are applied to them [11]. ScanObjectNN-PB_T50_RS contains 13698 real-world point cloud objects from 15 categories. 11,416 objects are used for training and 2,282 objects are used for testing. Each point cloud object has 2048 points, and their coordinates are not normalized. Each point in the point cloud has a point-level label to indicate the foreground or background. We assume that only S points are labeled within each real-world point cloud sample. Specifically, in the following experiments, we uniformly sample 1% points for each training sample as supervision.

7.3. Evaluation Metrics

For pseudo labels generation experiments, we calculate the segmentation accuracy for each sample and report the average segmentation accuracy over all instances (InsAvg) and all categories (CatAvg). The average refined segmentation accuracy

Table 1. The class-specific results of pseudo labels generated by PointRGCN with different levels of supervision.

Index	Model	Re-InsAvg	InsAvg	CatAvg	bag	bin	box	cabinet	chair	desk	display	door	shelf	table	bed	pillow	sink	sofa	toilet
Acc	GCN_1%	82.78	82.18	82.28	84.61	**82.66**	80.82	81.68	84.14	75.39	85.28	84.42	**77.41**	83.25	81.36	81.94	**82.13**	**82.35**	**86.79**
	PointRGCN_1%	**84.86**	**82.49**	**82.57**	**85.08**	82.32	**81.24**	**82.94**	**84.35**	**75.91**	**85.40**	**85.61**	76.66	**84.07**	**81.60**	**82.51**	81.98	82.31	86.53
	GCN_10%	87.70	85.69	85.81	88.25	86.43	84.88	85.30	87.63	79.35	88.49	87.31	80.80	86.48	84.91	85.71	85.43	86.19	90.05
	PointRGCN_10%	**94.44**	**92.20**	**92.42**	**94.12**	**92.76**	**92.75**	**92.63**	**93.59**	**88.20**	**94.18**	**93.84**	**86.38**	**92.37**	**92.09**	**93.38**	**92.82**	**92.35**	**94.81**
mIoU	GCN_1%	-	61.81	62.43	66.97	64.11	61.51	61.33	65.24	52.58	65.80	61.16	52.56	59.61	62.64	64.91	63.54	63.60	70.91
	PointRGCN_1%	-	**64.46**	**64.93**	**69.22**	**65.07**	**63.96**	**65.35**	**67.29**	**56.66**	**67.65**	**65.83**	**54.95**	**63.92**	**64.54**	**67.19**	**64.95**	**65.26**	**72.11**
	GCN_10%	-	69.88	70.51	75.68	72.58	70.10	69.73	73.24	60.65	73.31	68.77	60.28	67.56	70.64	72.86	71.09	72.20	78.94
	PointRGCN_10%	-	**83.13**	**83.79**	**87.32**	**84.72**	**85.27**	**84.02**	**85.33**	**77.62**	**86.24**	**83.96**	**72.17**	**81.34**	**84.33**	**87.05**	**85.00**	**84.09**	**88.38**

over all instances is also reported as ’Re-InsAvg’. At the same time, the mean Intersect over Union (mIoU) for each sample is calculated, and the average mIoU over all instances (InsAvg) and all categories (CatAvg) are also reported.

For real-world point cloud classification experiments, overall accuracy (oAcc) and mean class accuracy (mAcc) are used as performance criteria. ’oAcc’ represents the mean accuracy for all test instances and ’mAcc’ represents the mean accuracy for all shape classes [51]. Meanwhile, ’SegAcc’, the average segmentation accuracy over all samples is also used to evaluate the segmentation branch performance.

7.4. Pseudo Labels Generation Results

We use the proposed PointRGCN to generate pseudo labeled data for the incompletely labeled ScanObjectN-PB_T50_RS training set. In Table 1, we report the performance of GCN and the proposed PointRGCN. Our proposed method can correctly classify 92.2% of the total points with only 10% of the labeled points. Under the setting of scarce labeling, our method can still correctly classify about 81.4% of the total points even with only 1% of the labeled points. Meanwhile, the PointRGCN exceeds the baseline GCN employed on the *k*-NN graph at different labeling levels. Moreover, by the smoothness constraint module, the misclassification can be eliminated, and only the ”possibly correct” points can be retained, which further improves the accuracy.

7.5. Real-world Point Cloud Classification Results

The results of our classification framework on the ScanObjectNN-PB_T50_RS dataset are shown in Table 2. The baseline method indicates that only a few ground-truth point-level labels are used as supervision. By comparing the results of different models, we can draw the following conclusions. First of all, our proposed method based on pseudo labels generation and noisy label learning improve the oAcc by 0.7% compared with the baseline. Secondly, our weakly supervised real-world point cloud classification network(Ours) is comparable to the fully-supervised methods [11] even with only 1% point-level labels. The gap between the above two methods is only 0.3%. Finally, our method outperforms 1.5% improvement on overall accuracy compared with the DGCNN without a segmentation-guided branch.

7.6. Ablation Analysis

We further conduct detailed experiments to evaluate the importance of the proposed components, and examine the compatibility of the proposed WSC-Net with different backbone networks.

Table 2. Classification results on ScanObjectNN-PB_T50_RS.

Setting	Model	oAcc	mAcc
Only Shape-level Labels	3DmFV [52]	63.0	58.1
	PointNet [1]	68.2	63.4
	SpiderCNN [31]	73.7	69.8
	PointNet++ [6]	77.9	75.4
	DGCNN [7]	78.1	73.6
	PointCNN [8]	78.5	75.1
Fully Supervised	BGA−PN++ [11]	**80.2**	**77.5**
	BGA−DGCNN [11]	**79.9**	**75.7**
Weakly Supervised	Ours(1%baseline)	79.0	75.0
	Ours(1%)	79.7	75.9
	Ours(10%baseline)	79.9	76.6
	Ours10%	**79.9**	**76.6**

Table 3. Importance of different modules.

PS	SC	REG	NRL	oAcc	mAcc	SegAcc
				79.0	75.0	78.0
x				79.0	75.3	75.9
x	x			79.4	75.8	75.7
x	x	x		79.3	**76.3**	78.0
x	x	x	x	**79.7**	75.9	**78.3**

7.6.1. The importance of each component

In order to evaluate the importance of each component, we evaluate the performance of the combination of different components, and the results are summarized in Table 3. ”PS” denotes we use the generated pseudo labels as additional supervision for the segmentation branch. ”SC” denotes smoothness constraint module is employed to refine the pseudo labels. ”REG” denotes we assign weights to $\mathcal{L}_{weak}$ and $\mathcal{L}_{pseudo}$. ”NRL” denotes we introduce noise-robust loss to $\mathcal{L}_{pseudo}$.

As pseudo labels contain some misclassifications, it’s harder to train an accurate model through corrupted labels. Therefore, We observe that summing $\mathcal{L}_{weak}$ and $\mathcal{L}_{pseudo}$ indiscriminately to train the segmentation branch will not improve the classification model. Explicitly refining pseudo labels by using the smoothness constraint module leads to about 0.4% improvement for overall accuracy. By re-weighting $\mathcal{L}_{weak}$ and $\mathcal{L}_{pseudo}$, there is about 0.3% improvement compared with the baseline. Combining the above two strategies and applying noise-robust loss on pseudo label loss further improve the overall accuracy by 0.4%.

7.6.2. Compatibility

We further evaluate the compatibility with alternative backbone networks of our framework. Specifically, we conduct a compatibility experiment on PointNet++. As shown in Table 4, our method can achieve similar results under different networks. At the same time, we observe that the proposed method

Table 4. Compatibility with alternative backbone network.

Model	SegAcc	oAcc	mAcc
PointNet++ [6]	47.37	79.17	77.02
BAG_PointNet++ [11]	-	80.20	77.50
WS_PointNet++_1%_baseline	76.01	79.48	**77.67**
WS_PointNet++_1%	**78.45**	**80.07**	77.14
WS_PointNet++_10%_baseline	77.29	**80.24**	**77.88**
WS_PointNet++_10%	**78.56**	80.14	76.97

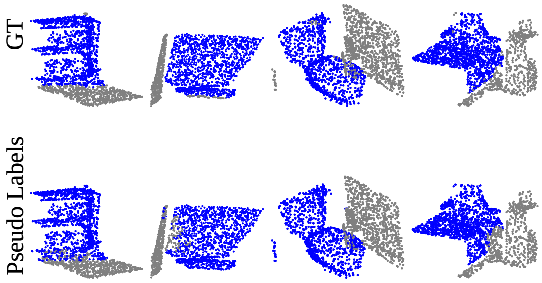


Fig. 7. 1% labeled point cloud pseudo labels generation resutls.

has a significant improvement in segmentation accuracy compared with the baseline. However, under the 10% labeled setting, the network focuses too much on the segmentation task, and the performance on the classification task drops.

7.7. *Visualization*

We show some qualitative results of real-world point cloud object foreground-background predictions in both the pseudo labels generation on training set and the final output segmentation results on test set. For each segmentation result, background and foreground are labeled in gray and blue, respectively.

Firstly, we present the pseudo labels generation results on scanobjectNN training set samples in Fig. 7. For each sample, the top row is the ground-truth label, and the bottom row is the pseudo labels generated with only 1% point-level labels. It can be observed that our proposed R-GCN based method can generate very accurate pseudo labels with a small amount of point-level labels. Nonetheless, we also observe that our method misclassifies some outliers.

Then, we visualize the foreground-background segmentation results of the auxiliary segmentation branch. For each sample, the top row is the ground-truth annotations, and the bottom row is the foreground-background predictions of our model trained with 1% ground-truth labels. Fig.8 and Fig.9 are the foreground-background segmentation results of correctly classified samples, and the foreground and background segmentation results of incorrectly classified samples,respectively. It can be observed that, for correctly classified samples, its segmentation results are relatively accurate. While for incorrectly classified samples, its segmentation results are relatively bad.

8. Conclusion

In this paper, we propose a weakly supervised classification framework, which requires only sparse point-level foreground-background annotations, for classifying real-world point cloud objects. The proposed pseudo labels generation method PointRGCN can significantly reduce the labor and time costs of annotating 3D real-world datasets. Besides, the introduced noise-robust multi-task loss can improve the robustness against noisy foreground-background labels. Experiments on the ScanobjectNN dataset show that our framework is comparable with many popular or state-of-the-art fully-supervised methods with even only 1% point-level labels.

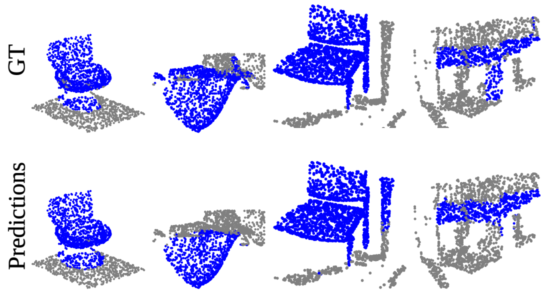


Fig. 8. Correctly classified samples segmentation results.

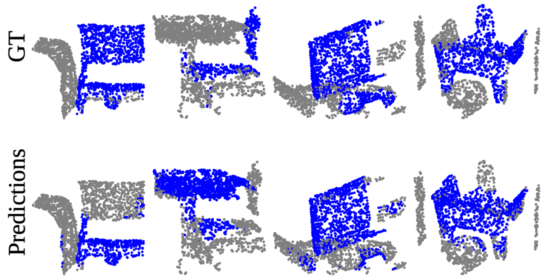


Fig. 9. Incorrectly classified samples segmentation results.

Acknowledgments

The authors would like to acknowledge the helpful comments and kindly suggestions provided by anonymous referees.

References

[1] Qi, CR, Su, H, Mo, K, Guibas, LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 652–660.

[2] Xu, M, Zhou, Z, Qiao, Y. Geometry sharing network for 3d point cloud classification and segmentation. In: Proceedings of the Thirty-Three AAAI Conference on Artificial Intelligence. 2020, p. 12500–12507.

[3] Ku, T, Galanakis, S, Boom, B, Veltkamp, RC, Bangera, D, Gangisetty, S, et al. Shrec 2021: 3d point cloud change detection for street scenes. Computers & Graphics 2021;99:192–200.

[4] Wang, W, Su, T, Liu, H, Li, X, Jia, Z, Zhou, L, et al. Surface reconstruction from unoriented point clouds by a new triangle selection strategy. Computers & Graphics 2019;84:144–159.

[5] Guo, Y, Wang, H, Hu, Q, Liu, H, Liu, L, Bennamoun, M. Deep learning for 3d point clouds: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 2020;:1–1doi:**10.1109/TPAMI.2020.3005434**.

[6] Qi, CR, Yi, L, Su, H, Guibas, LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. 2017, p. 5099–5108.

[7] Wang, Y, Sun, Y, Liu, Z, Sarma, SE, Bronstein, MM, Solomon, J. Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics 2019;38(5):146.

[8] Li, Y, Bu, R, Sun, M, Wu, W, Di, X, Chen, B. Pointcnn: Convolution on x-transformed points. In: Advances in neural information processing systems. 2018, p. 820–830.

[9] Mao, J, Wang, X, Li, H. Interpolated convolutional networks for 3d point cloud understanding. In: Proceedings of the IEEE International Conference on Computer Vision. 2019, p. 1578–1587.

[10] Thomas, H, Qi, CR, Deschaud, JE, Marcotegui, B, Goulette, F, Guibas, LJ. Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE International Conference on Computer Vision. 2019, p. 6411–6420.

[11] Uy, MA, Pham, QH, Hua, BS, Nguyen, T, Yeung, SK. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE International Conference on Computer Vision. 2019, p. 1588–1597.

[12] Alliegro, A, Boscaini, D, Tommasi, T. Joint supervised and self-supervised learning for 3d real-world challenges. arXiv preprint arXiv:200407392 2020;.

[13] Fuchs, F, Worrall, D, Fischer, V, Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. Advances in Neural Information Processing Systems 2020;33.

[14] Zhao, C, Yang, J, Xiong, X, Zhu, A, Cao, Z, Li, X. Rotation invariant point cloud classification: Where local geometry meets global topology. arXiv preprint arXiv:191100195 2019;.

[15] Mo, K, Zhu, S, Chang, AX, Yi, L, Tripathi, S, Guibas, LJ, et al. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, p. 909–918.

[16] Armeni, I, Sener, O, Zamir, AR, Jiang, H, Brilakis, I, Fischer, M, et al. 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, p. 1534–1543.

[17] Zhou, ZH. A brief introduction to weakly supervised learning. National Science Review 2018;5(1):44–53.

[18] Su, H, Maji, S, Kalogerakis, E, Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 945–953.

[19] Feng, Y, Zhang, Z, Zhao, X, Ji, R, Gao, Y. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 264–272.

[20] Ma, C, Guo, Y, Yang, J, An, W. Learning multi-view representation with lstm for 3-d shape recognition and retrieval. IEEE Transactions on Multimedia 2018;21(5):1169–1182.

[21] Wu, Z, Song, S, Khosla, A, Yu, F, Zhang, L, Tang, X, et al. 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, p. 1912–1920.

[22] Maturana, D, Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2015, p. 922–928.

[23] Li, J, Chen, BM, Hee Lee, G. So-net: Self-organizing network for point cloud analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 9397–9406.

[24] Simonovsky, M, Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 3693–3702.

[25] Zhang, Y, Rabbat, M. A graph-cnn for 3d point cloud classification. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2018, p. 6279–6283.

[26] Wang, C, Samari, B, Siddiqi, K. Local spectral graph convolution for point set feature learning. In: Proceedings of the European conference on computer vision (ECCV). 2018, p. 52–66.

[27] Landrieu, L, Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 4558–4567.

[28] Te, G, Hu, W, Zheng, A, Guo, Z. Rgcnn: Regularized graph cnn for point cloud segmentation. In: Proceedings of the 26th ACM international conference on Multimedia. 2018, p. 746–754.

[29] Li, G, Muller, M, Thabet, A, Ghanem, B. Deepgcns: Can gcns go as deep as cnns? In: Proceedings of the IEEE International Conference on Computer Vision. 2019, p. 9267–9276.

[30] Wu, W, Qi, Z, Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, p. 9621–9630.

[31] Xu, Y, Fan, T, Xu, M, Zeng, L, Qiao, Y. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018, p. 87–102.

[32] Hua, BS, Tran, MK, Yeung, SK. Pointwise convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 984–993.

[33] Li, X, Yu, L, Fu, CW, Cohen-Or, D, Heng, PA. Unsupervised detection of distinctive regions on 3d shapes. ACM Transactions on Graphics (TOG) 2020;39(5):1–14.

[34] Thabet, A, Alwassel, H, Ghanem, B. Self-supervised learning of local features in 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. 2020,.

[35] Sauder, J, Sievers, B. Self-supervised deep learning on point clouds by reconstructing space. In: Advances in Neural Information Processing Systems. 2019, p. 12962–12972.

[36] Xu, X, Lee, GH. Weakly supervised semantic point cloud segmentation:towards 10x fewer labels. arXiv preprint arXiv:200404091 2020;.

[37] Wei, J, Lin, G, Yap, KH, Hung, TY, Xie, L. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, p. 4384–4393.

[38] Vahdat, A. Toward robustness against label noise in training deep discriminative neural networks. In: Advances in Neural Information Processing Systems. 2017, p. 5596–5605.

[39] Lee, KH, He, X, Zhang, L, Yang, L. Cleannet: Transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 5447–5456.

[40] Li, Y, Yang, J, Song, Y, Cao, L, Luo, J, Li, LJ. Learning from noisy labels with distillation. In: Proceedings of the IEEE International Conference on Computer Vision. 2017, p. 1910–1918.

[41] Jiang, L, Zhou, Z, Leung, T, Li, LJ, Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning. 2018, p. 2304–2313.

[42] Han, B, Yao, Q, Yu, X, Niu, G, Xu, M, Hu, W, et al. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: Advances in neural information processing systems. 2018, p. 8527–8537.

[43] Tanaka, D, Ikami, D, Yamasaki, T, Aizawa, K. Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, p. 5552–5560.

[44] Han, B, Yao, J, Niu, G, Zhou, M, Tsang, I, Zhang, Y, et al. Masking: A new perspective of noisy supervision. In: Advances in Neural Information Processing Systems. 2018, p. 5836–5846.

[45] Ma, X, Huang, H, Wang, Y, Romano, S, Erfani, S, Bailey, J. Normalized loss functions for deep learning with noisy labels. In: Proceedings of the International Conference on Machine Learning 1 pre-proceedings (ICML 2020). 2020,.

[46] Ghosh, A, Kumar, H, Sastry, P. Robust loss functions under label noise for deep neural networks. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. 2017, p. 1919–1925.

[47] Zhang, Z, Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In: Advances in neural information processing systems. 2018, p. 8778–8788.

[48] Wang, Y, Ma, X, Chen, Z, Luo, Y, Yi, J, Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In: IEEE International Conference on Computer Vision. 2019,.

[49] Schlichtkrull, MS, Kipf, T, Bloem, P, Den Berg, RV, Titov, I, Welling, M. Modeling relational data with graph convolutional networks. European Semantic Web Conference (2018) 2018;:593–607.

[50] Lee, DH. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML; vol. 3. 2013,.

[51] Guo, Y, Wang, H, Hu, Q, Liu, H, Liu, L, Bennamoun, M. Deep learning for 3d point clouds: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 2020;.

[52] Ben-Shabat, Y, Lindenbaum, M, Fischer, A. 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. IEEE Robotics and Automation Letters 2018;3(4):3145–3152.