

Collaborative Propagation on Multiple Instance Graphs for 3D Instance Segmentation with Single-point Supervision

Shichao Dong^{1,2} Ruibo Li^{1,2} Jiacheng Wei² Fayao Liu³ Guosheng Lin^{1,2}*

¹ S-lab, Nanyang Technological University, Singapore

²School of Computer Science and Engineering, Nanyang Technological University, Singapore

³ Institute for Infocomm Research, A*STAR, Singapore

{scdong, gslin}@ntu.edu.sg {ruibo001, jiacheng002}@e.ntu.edu.sg fayao.liu@gmail.com

Abstract

Instance segmentation on 3D point clouds has been attracting increasing attention due to its wide applications, especially in scene understanding areas. However, most existing methods operate on fully annotated data while manually preparing ground-truth labels at point-level is very cumbersome and labor-intensive. To address this issue, we propose a novel weakly supervised method **RWSeg** that only requires labeling one object with one point. With these sparse weak labels, we introduce a unified framework with two branches to propagate semantic and instance information respectively to unknown regions using self-attention and a cross-graph random walk method. Specifically, we propose a Cross-graph Competing Random Walks (CRW) algorithm that encourages competition among different instance graphs to resolve ambiguities in closely placed objects, improving instance assignment accuracy. RWSeg generates high-quality instance-level pseudo labels. Experimental results on ScanNet-v2 and S3DIS datasets show that our approach achieves comparable performance with fully-supervised methods and outperforms previous weakly-supervised methods by a substantial margin.

1. Introduction

With the rapid development of 3D sensing technology, point cloud based scene understanding has become a popular research topic in recent years. Instance segmentation is one of the most fundamental tasks in this field and has many applications in robotics, autonomous driving, AR/VR, etc. Given a 3D point cloud depicting a scene, this task requires predicting not only a semantic category but also an instance id to differentiate objects at point level. Many deep learning methods have been developed for this task, showing promising results. However, most of these methods operate

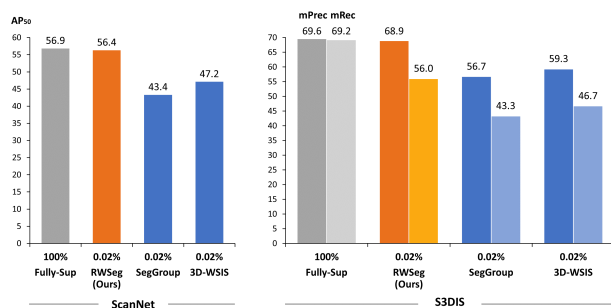


Figure 1. Comparisons of our approach RWSeg with two recent weakly supervised 3D instance segmentation methods and the fully-supervised baseline on two datasets. Our method achieves better results than other weakly supervised methods with the same amount of weak annotations.

on point-wise fully annotated data to supervise the network training.

Manually creating data annotations at point level is very cumbersome and labor-intensive. Although some tools have been adopted to assist, the average time used to annotate one scene is about 22.3 minutes on ScanNet-v2 dataset [10]. To alleviate this issue, several types of weak annotations have been proposed, such as scene-level annotation, subcloud-level annotation [47], 2D image based annotation and 3D bounding box annotation [1, 8]. However, not all weak label types are easy to obtain in practice. In this work, we adopt the annotation method used in SegGroup [40] and “One Thing One Click” [33], which only requires annotating a single point for each object. As shown in Figure 2, this results in very sparse initial annotations, with less than 0.02% of total points requiring labeling. According to [40, 33], this annotation method takes less than two minutes per scene, significantly reducing the need for human effort.

Tao et al. [40] and Tang et al. [39] have investigated the “One-thing-one-click” approach to address the challenge of weakly supervised 3D instance segmentation. These techniques construct graphs on top of the over-segmentation

*Corresponding author: G.Lin (e-mail:gslin@ntu.edu.sg)

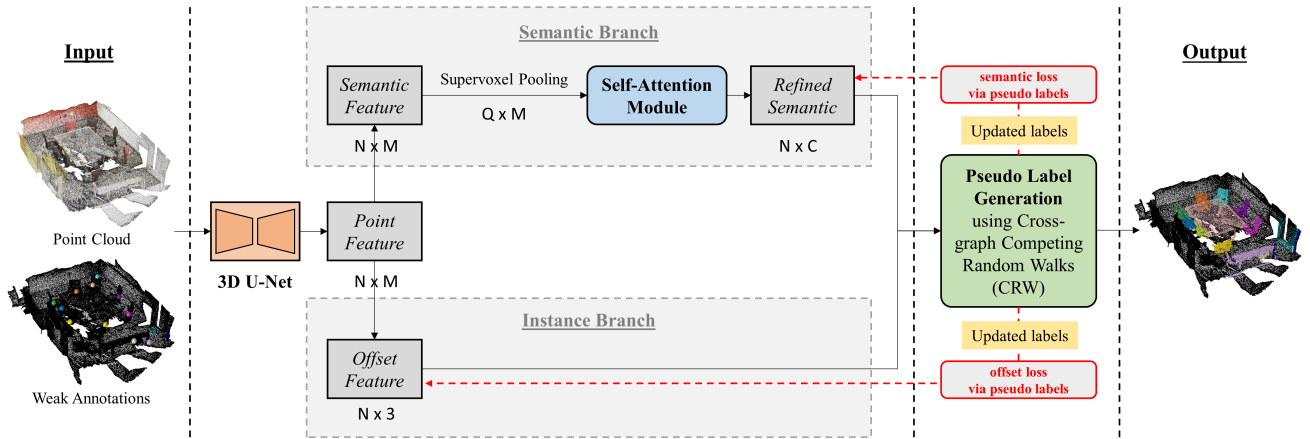


Figure 2. Pipeline of our proposed weakly supervised method for 3D instance segmentation. The input point cloud is annotated with a single point for each object (enlarged for better visualizations). We use a 3D U-Net backbone based on submanifold sparse convolution [17] to extract point features. Next, we apply average pooling to the points within the same supervoxel. To facilitate semantic feature propagation, we utilize a self-attention module. Finally, our novel Cross-graph Competing Random Walks (CRW) module leverages the inputs from both branches to generate high-quality pseudo labels for further network training.

outcomes and apply Graph Convolution Network (GCN) or inter-superpoint affinity for label propagation. However, these approaches encounter some issues. SegGroup [40] relies solely on a cross-entropy loss for its semantic prediction with a greedy algorithm for clustering, hence lacking instance-related information. Besides, this method is only designed for the purpose of generating pseudo labels, and therefore requires to utilize these pseudo labels as ground-truth to train a separate network for prediction. 3D-WSIS [39] utilizes an offset loss and an affinity loss to produce better discriminative features, but their graph is based on the over-segmented point clouds, and the feature of each supervoxel is simply obtained through average pooling of point features and coordinates. The size of supervoxels can vary significantly in their setup, and the initial weak labels can be located at any part of objects, resulting in an unbalanced attraction to neighboring nodes. This may lead to difficulty in identifying precise boundaries, particularly when multiple instances are located close to each other.

In this paper, we propose a novel weakly supervised learning approach, named **RWSeg**, for 3D point cloud instance segmentation. With only one point annotation per instance, we focus on two key considerations: (1) effective feature propagation is critical for generating high-quality pseudo labels, and (2) leveraging the interactions among instance graphs can be beneficial in finding more accurate instance boundaries and improving the quality of clustering. To address the limitations of previous methods, we are motivated to develop a unified structure for both feature learning and feature propagation.

Convolutional Neural Network (CNN) can extract good local features. However, long-range dependencies can hardly be captured due to its relatively small receptive field.

The limitations of CNNs in capturing long-range dependencies are exacerbated in weakly supervised learning scenarios, where only a limited number of certain labels are available to supervise the training process. To this end, we introduce a self-attention module after the 3D CNN backbone, which can effectively propagate long-range information to unknown regions.

For instance pseudo label generation, a customized random walk algorithm on point-level is developed for 3D weakly instance segmentation. The point clouds are first split by their categories, and for each category, multiple instance graphs are built and random walk propagation is performed on each of them. The total energy on each individual graph is identical, based on the assumption that same-class objects tend to have similar sizes. A competing mechanism is designed to perform collaborative propagation on multiple instance graphs. To sum up, the key contributions of our work are as follows:

- We design a unified framework for weakly supervised 3D instance segmentation. To enhance the feature propagation, we introduce a self-attention module to capture long-range dependencies.
- We propose a novel algorithm to perform collaborative propagation on multiple instance graphs to generate high-quality instance pseudo labels. The designed competing mechanism helps to resolve ambiguous cases in 3D instance segmentation task.
- With significantly fewer annotations, our method bridges the gap between weakly supervised learning and fully supervised learning in 3D instance segmentation.

2. Related Work

Fully supervised 3D segmentation To effectively process unstructured and unordered 3D data, current feature learning methods can be broadly categorized into two types: point-based methods [28, 37, 38, 41, 49, 53, 13, 18] and voxel-based methods [16, 17, 23, 27]. Voxel-based approaches involve transforming data into 3D volumetric grids, whereas point-based methods operate directly on the individual points. Instance segmentation on point clouds can be seen as a joint task of segmentation and localization. Proposal-based methods [45, 24, 51, 14] detect object boundaries explicitly and then perform binary mask segmentation as the final output. On the other hand, proposal-free methods [45, 32, 46, 35, 26, 25, 19, 6, 29, 12] directly regress instance centroids without performing the detection task. Jiang et al. [25] utilized a submanifold sparse convolution [17] based 3D U-net and proposed to use a breadth-first search clustering algorithm on dual coordinate sets.

Weakly supervised segmentation Numerous weakly supervised methods have been proposed for image segmentation [3, 22, 36, 34, 2, 55, 5]. Wei et al. [48] proposed the first weakly supervised approach for point cloud semantic segmentation, utilizing Class Activation Map (CAM) to generate point-level pseudo labels with subcloud-level annotations. Several subsequent works [50, 44, 33, 11] also addressed weakly segmentation on point clouds with lesser supervision. There have been limited attempts to solve 3D weakly supervised instance segmentation. Hou et al. [21] designed a pre-training method to assist prediction, while Tao et al. [40] proposed Seggroup with graph convolution network (GCN) for instance label propagation. However, Seggroup lacks the ability to learn discriminative features for separating instances. Tang et al. [39] proposed to learn discriminative features and use inter-superpoint affinity for label propagation. However, their method did not fully utilize all the spatial information and may affect their performance on ambiguous cases. Liao et al. [1] and J. Chibane et al. [8] proposed using 3D bounding boxes as supervision. However, box annotation provides much richer information than clicking one point per instance, and most non-overlapped objects can already be defined by 3D bounding box. This may lessen the significance of their work.

3. Method

In this section, we first introduce our data annotation setting for point cloud instance segmentation in Section 3.1. Then, Section 3.2 describes our training strategy. Lastly, Section 3.3 and 3.4 present our approach in detail, including network architecture, semantic branch, instance branch and proposed pseudo label generation algorithm.

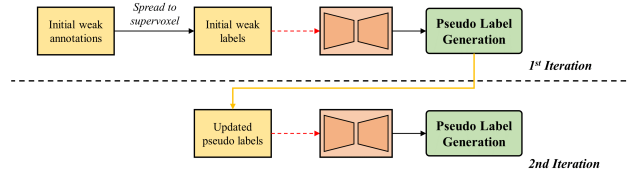


Figure 3. Learning cycle of our proposed weakly supervised method for 3D Instance Segmentation.

3.1. Weak Annotation

Following SegGroup [40], we adopt the annotation setting of one point per object, as shown in Figure 2. To create initial pseudo labels, we spread the labels from annotated points to nearby points within the same supervoxel segment. These segments are generated by unsupervised over-segmentation method [15] based on the surface normals of points. Points within the same segment have high internal consistency, which are used as the initial ground-truth to supervise the network training.

3.2. Learning Strategy

As shown in Figure 3, the network training of our method consists of two stages. The first stage is supervised by the initial weak labels. Afterward, predictions with high confidence from our pseudo label generation algorithm are further updated as new ground-truth labels for next stage training. With this learning strategy, the quality of learned features can be consistently improved

3.3. Network Architecture

Our network takes point cloud $P \in \mathbb{R}^{N \times 3}$ as input where N is the number of points in P . It uses a shared U-Net backbone and two separate branches for point-level semantic feature learning and instance centroid regression. In the semantic branch, a self-attention based module is used to further enhance semantic features, especially for those regions without supervision. Following that, our proposed **Cross-Graph Competing Random Walks (CRW)** algorithm leverages learned features and existing ground-truth weak labels to generate instance-level pseudo labels. With refined weak labels, the network can be further trained to produce better features. All proposed modules are within the unified framework, as shown in Figure 2.

Semantic segmentation branch The submanifold sparse convolution [17] based backbone network can extract point-wise features with good local information capturing. However, to enhance the network’s ability to capture long-range feature dependencies and extend its receptive field, we propose incorporating a self-attention module to further refine the semantic features. In order to reduce the computational complexity of self-attention and ensure local geometric consistency, we utilize a supervoxel generation method

[31]. Specifically, for each supervoxel set, we apply average pooling to both point coordinates and semantic features. Following [42, 54], we build a self-attention layer across all the supervoxels and then interpolate the output to the original size of the input point cloud. During training, we use a conventional cross-entropy loss H_{CE} with incomplete labels to supervise the process. The structure diagram and formulas of the self-attention module are provided in the supplementary.

Instance centroid offset branch Parallel to the semantic branch, we apply a 2-layers MLP upon point features to predict point-wise centroid shift vector $d_i \in \mathbb{R}^3$. The instance centroid \hat{q} is defined as the mean coordinates of all points with the same instance label. Following [25], We use an L_1 regression loss and a cosine similarity based direction loss to train the offset prediction. We only consider foreground points with weak labels or pseudo weak labels for supervision. With initial weak labels, real centroids of instances can hardly be inferred. However, we found it is still beneficial to apply offset loss, as it can help to slightly shift points towards inner part of objects.

The final joint loss function can be written as

$$L_{joint} = L_{sem} + L_{offset}. \quad (1)$$

3.4. Pseudo label Generation

After training with the initial weak labels, we now have a network that can make semantic prediction and offset prediction, which can be further utilized to generate pseudo instance labels. However, due to the limited supervision used during model training, the quality of the prediction may not be very accurate at the first iteration. To address this issue, we propose a random walk-based algorithm to generate reliable pseudo labels for unlabelled points.

In this section, we first describe how we construct an individual graph in Figure 4 and then present the details of cross-graph competing mechanism and the clustering algorithm in Figure 5. The core idea of our algorithm is to enable interactions among instance graphs and gradually updates seeding points until reaching a signal equilibrium state.

Building graph on the point cloud According to the semantic predictions from the semantic branch $\mathbf{S} = \{s_1, s_2, \dots, s_N\} \in \mathbb{R}^N$ on point clouds \mathbf{P} , we treat each foreground semantic category as a target group. For each group, we build K fully connected and undirected instance graphs, with K being the number of instances. As shown in Figure 4, the nodes of each graph are points from all K instances. Each node in each graph is associated with an initial binary label (score), as detailed in the paragraph below. The K instance graphs have the same nodes and edges, with

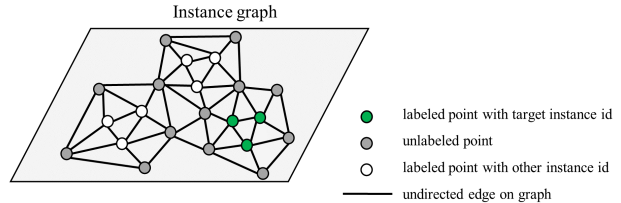


Figure 4. Illustration of a single instance graph. The nodes on graph are connected by undirected edges. The edge weights are determined by the transition matrix \mathbf{A} in Eq. (5). The initial node values are determined by the vector \mathbf{b}_0 in Eq. (2) For this example, the initial values of the three green nodes are $1/3$.

the only difference being they have different initial graph node score vectors.

For the l -th instance graph, its initial graph node score vector \mathbf{b}_0^l is defined by its binary instance label mask \mathbf{m}^l , with the i -th element $m_i^l = 1$ if the i -th node (point) has an instance label of l . The i -th element of \mathbf{b}_0^l is:

$$b_0^{l(i)} = \begin{cases} \frac{1}{\sum_{j=0}^n m_j^l} & m_i^l = 1 \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where n is the number of nodes in the graph. The process of normalizing seeding points' initial scores involves dividing them by the total number of nodes corresponding to the same instance id. This normalization aims to achieve equitable allocation of the initial potential among instance graphs, thereby preventing any undue advantage for instances having a larger number of positive weak labels.

The random walk operation on each graph can be modeled with an $n \times n$ transition matrix \mathbf{A} . $\mathbf{A}_{ij} \in [0, 1]$ denotes the transition probability between i -th and j -th nodes, with a higher value indicating a higher transition probability.

To build transition matrix \mathbf{A} , we first consider a pairwise kernel function to derive a symmetric affinity matrix \mathbf{W} , which helps to enhance local smoothness. For each edge connecting the i -th and j -th nodes, we define its weight as:

$$\mathbf{W}_{ij} = \exp\left(-\frac{\|(x_i + d_i) - (x_j + d_j)\|^2}{2\sigma^2}\right), \quad (3)$$

where σ is a hyperparameter, x_i and x_j are point coordinates. We use the predicted offset vector \mathbf{d} from the instance branch to shift points from their original coordinates toward their instance centers. Node pairs with small euclidean distances in the 3D space tend to have high similarities.

Next, we formulate the transition matrix \mathbf{A} by the following rules. Specifically, we assign a weight of zero to the edges connecting nodes belonging to different instance labels, in order to restrict the direct interaction between them.

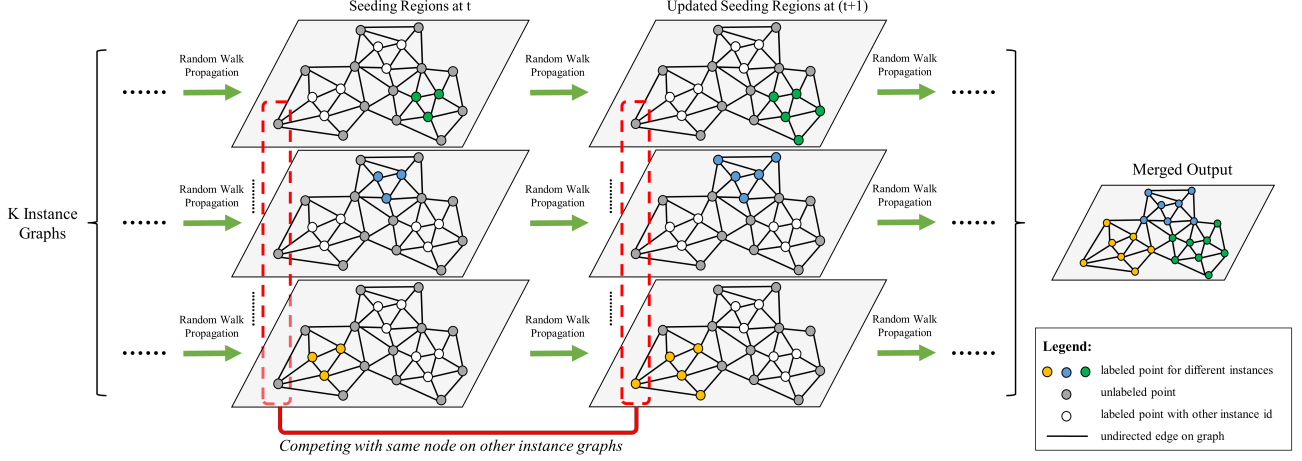


Figure 5. Illustration on Cross-graph Competing Random Walks (CRW). Our algorithm takes a group of points from the same semantic category as input and constructs K graphs according to the number of instances. Proposed method enables the interactions among the same positioned nodes on K instance graphs. Point score can be suppressed or enhanced after cross-graph competition, thereby affecting the following seeding points update strategy. High score points enjoy the priority to be grouped first. After performing several iterations, instance graphs are merged to generate the final output prediction.

$$\mathbf{A}_{ij} = \begin{cases} 0 & \widehat{L}_i \neq \widehat{L}_j \\ \mathbf{W}_{ij} & \text{otherwise} \end{cases}, \quad (4)$$

where \widehat{L}_i and \widehat{L}_j are the instance labels of two nodes. Lastly, transition matrix \mathbf{A} needs to be normalized:

$$\mathbf{A}_{ij} = \frac{\mathbf{A}_{ij}}{\sum_{j \in n} \mathbf{A}_{ij}}. \quad (5)$$

This transition matrix \mathbf{A} is shared among each group of instance graphs.

Random walk algorithm is performed by repeatedly adjusting node vector \mathbf{b} via the transition matrix \mathbf{A} . At t -th iteration, the adjustment can be expressed as

$$\mathbf{b}_{t+1}^l = \alpha \mathbf{A} \mathbf{b}_t^l + (1 - \alpha) \mathbf{b}_0^l, \quad (6)$$

where $\alpha \in [0, 1]$ is a blending coefficient between propagated scores and the initial scores.

When repeatedly applying unlimited random steps on a graph, it will reach equilibrium. The final steady-state of random walk algorithm can be written as

$$\mathbf{b}_{(\infty)}^l = (1 - \alpha)(\mathbf{I} - \alpha \mathbf{A})^{-1} \mathbf{b}_0^l. \quad (7)$$

Cross-graph Competing Random Walks (CRW) On top of the random walk algorithm, we design a mechanism to encourage competitive interactions among instance graphs, in Figure 5. Intuitively, the idea is to suppress a point's activation score in the current graph if its scores in other instance graphs are relatively high. However, the level of repulsive effect needs to be well controlled. Otherwise,

Algorithm 1 Cross-graph Competing Random Walks (CRW)

Input: coordinates $\mathbf{X} = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^{N \times 3}$; number of instances per category $K = \{k_1, k_2, \dots, k_s\}$ (s is the total number of valid classes); hyperparameter α, θ ; max iteration number t_{1max} and t_{2max} ; instance weak labels \widehat{L} ; semantic prediction \mathbf{S} ; offset prediction \mathbf{D}
Output: Instance pseudo label prediction \mathbf{P}

```

1: for  $id$  in foreground semantic IDs do
2:   for  $\mathbf{S} \in id$  do
3:     build  $K$  instance graphs ;
4:     construct affinity matrix  $\mathbf{W}$  via Eq. (3);
5:     construct transition matrix  $\mathbf{A}$  via Eq. (4);
6:     normalize transition matrix  $\mathbf{A}$  via Eq. (5);
7:     for  $l = 1$  to  $K$  do
8:       initialize graph node vector via Eq. (2);
9:       while  $t_1 \leq t_{1max}$  do
10:        for  $l = 1$  to  $K$  do
11:          propagate one step via Eq. (6);
12:           $t_1 \leftarrow t_1 + 1$ 
13:        while  $t_2 \leq t_{2max}$  do
14:          adjust node vectors via Eq. (8)
15:          for  $l = 1$  to  $K$  do
16:            reinitialize vector via Eq. (2);
17:            update top  $\theta$  as new seeding points
18:            propagate one step via Eq. (6);
19:             $t_2 \leftarrow t_2 + 1$ 
20:           $p_i \leftarrow \operatorname{argmax}(\mathbf{b}^{(i)})$ 
21:           $p_i \leftarrow \widehat{L}_j$  if under the same mask
22: return  $\mathbf{P}$ 

```

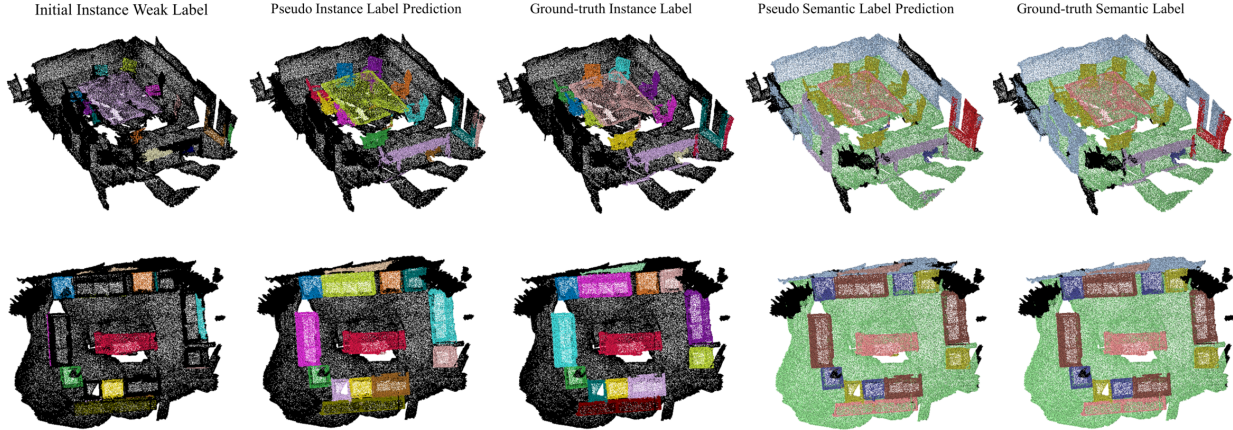


Figure 6. The qualitative visualization results of generated pseudo labels on ScanNet-v2 dataset[10]

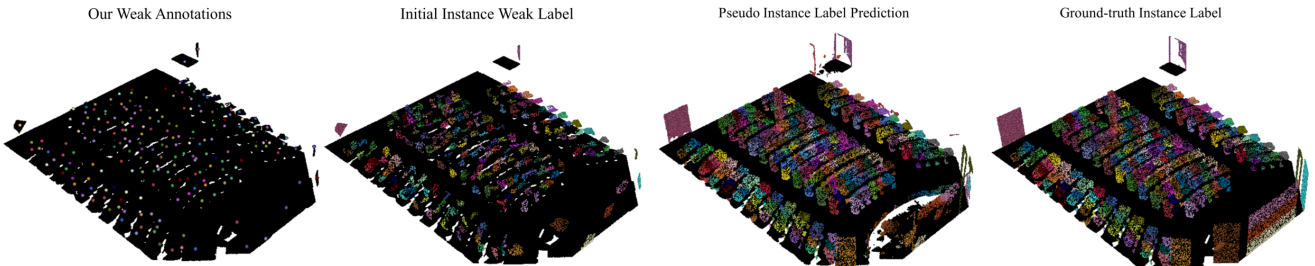


Figure 7. The qualitative visualization results of generated pseudo labels on S3DIS dataset [4]

too strong repulsive effects are likely to distort the results from the random walk.

Based on the random walk results, we apply a softmax function to every node score to adjust the probability distribution over the K instance graphs. Elements in the score vector are re-scaled to the range of $[0, 1]$, and the score values of the same positioned nodes on K instance graphs are summed to 1.

$$b^{l(i)} = \frac{\exp(b^{l(i)})}{\sum_{j=1}^K \exp(b^{j(i)})}, \quad (8)$$

where $b^{l(i)}$ denotes the score of the i -th node on l -th graph. In this simple manner, we bring repulsive interaction among instance graphs. A point that receives less competition from other instance graphs will be adjusted to a relatively higher score, and vice versa.

Then, for each instance, we pick a fixed percentage θ (i.e. 50%) of newly predicted pseudo labels with high confidence to be updated as seeding points for the next iteration. The selection is based on the sorted node scores. Only unlabelled points can be considered as new seeding points. Our approach gradually groups relatively confident points into seeds and performs a random walk step at each iteration.

4. Experiments

Datasets In this section, we show our experimental results on two public datasets: ScanNet-v2 [10] and S3DIS [4] to show the effectiveness of our proposed method. ScanNet-v2 dataset [10] is a popular 3D indoor dataset containing 2.5 million RGB-D views in 1513 real-world scenes, covering 20 semantic categories. The evaluation metrics of 3D instance segmentation are mean average precisions at different overlap percentages, i.e., mAP@0.25, mAP@0.5 and mAP respectively. S3DIS dataset [4] has 272 scenes under six large-scale indoor areas. Unlike ScanNet [10], all 13 classes including background are annotated as instances and require prediction. We use the mean precision (mPre) and mean recall (mRec) with an IoU threshold of 0.5 as the evaluation metric.

Implementation details We set the voxel size as $2cm$ for submanifold sparse convolution [17] based backbone, following [25]. Our network is trained on a single GPU card. For each stage of training, the backbone network and self-attention module are trained sequentially, with a batch size of 4 and 2 respectively. We set γ and δ in the self-attention module as two-layer MLPs with the hidden dimension of 64 and 32 respectively. For CRW algorithm, we set hyperparam-

Semantic mIoU	Label	wall	floor	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridge	showr	toil	sink	bath	ofurn	avg
MPRM [48]	Scene	47.3	41.1	10.4	43.2	25.2	43.1	21.9	9.8	12.3	45.0	9.0	13.9	21.1	40.9	1.8	29.4	14.3	9.2	39.9	10.0	24.4
MPRM [48]	Subcloud	58.0	57.3	33.2	71.8	50.4	69.8	47.9	42.1	44.9	73.8	28.0	21.5	49.5	72.0	38.8	44.1	42.4	20.0	48.7	34.4	47.4
SegGroup [40]	0.02%	71.0	82.5	63.0	52.3	72.7	61.2	65.1	66.7	55.9	46.3	42.7	50.9	50.6	67.9	67.3	70.3	70.7	53.1	54.5	63.7	61.4
RWSeg (Ours)	0.02%	88.8	94.4	80.2	82.4	85.9	91.2	76.5	76.6	78.2	87.5	66.3	64.1	67.7	85.6	86.9	88.9	92.4	71.5	91.7	75.3	81.6

Table 1. Pseudo label quality of semantic segmentation (category-level) on ScanNet-2 [10] training set.

Instance AP	Metric	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridge	showr	toil	sink	bath	ofurn	avg
RWSeg (Ours)	AP	59.0	65.9	70.3	82.1	59.3	38.2	54.0	68.0	54.7	35.6	35.8	48.0	73.9	80.6	88.4	44.6	85.4	54.3	61.0
	AP ₅₀	85.7	94.2	93.8	90.3	87.1	60.8	77.1	84.5	81.6	79.9	74.1	69.7	92.1	92.4	97.8	82.3	97.4	77.6	84.4
	AP ₂₅	96.4	99.0	98.3	95.7	95.2	87.0	91.6	91.5	92.8	96.3	93.9	87.5	99.2	97.4	99.3	95.8	100.0	92.8	95.0

Table 2. Pseudo label quality of instance segmentation on ScanNet-2 [10] training set.

eters $\alpha = 0.2$, $t_{1max} = 1$, $t_{2max} = 5$ and θ as 50%. Due to GPU memory limit, we subsample the input point cloud to CRW if the point number is above 25k. Last output remains at original resolution. For network training, we use Adam solver for optimization with an initial learning rate of 0.001.

Pseudo label evaluation As shown in Table 1 and Table 2, we present the quality of our generated pseudo labels based. Reported final pseudo labels are created after two stages of network training. Our network is trained only on the training set of ScanNet-v2 [10] with 1201 scenes, no extra data is needed. In Table 1, the semantic quality of our pseudo labels largely outperforms previous methods by at least 20.2%. Besides, we also report the instance quality of pseudo labels in Table 2. However, no available data from other methods can be used for comparison at present. Our qualitative pseudo labels can be used by any fully supervised method to resolve their annotation cost issue.

Prediction evaluation Different from weakly supervised methods like SegGroup [40] that require training another a new network for prediction, we can directly adopt other methods on the same network for prediction without retraining. Here we employ a Breadth-First Search (BFS) clustering algorithm from PointGroup [25] to our network. In Table 3, we compare the prediction results with fully supervised PointGroup [25] and other weakly supervised methods on ScanNet-v2 [10] validation set.

Our method significantly outperforms SegGroup [40] and 3D-WSIS [39] over all evaluation metrics, generally **by an absolute margin of around 10 points**. Remarkably, with only 0.02% of annotated points, we **achieve comparable results with fully supervised method [25]**. We also report the instance segmentation results on ScanNet-v2 [10] test set in Table 4. Our method again performs significantly better than other weakly supervised methods which use the same amount of annotations. For S3DIS [4] dataset, we report Area 5 and 6-fold cross validation results in Table 5.

Method	Supervision	AP	AP ₅₀	AP ₂₅
Full Supervision: PointGroup [25]	100%	34.8	56.9	71.3
Init+Act. Point Supervision: CSC-20 (PointGroup) [21]	20 pts/scene	-	27.2	-
CSC-50 (PointGroup) [21]	50 pts/scene	-	35.7	-
SPIB [1]	100% Box	-	38.6	61.4
Box2Mask [8]	100% Box	-	59.7	71.8
TWIST [9]	1%	9.6	17.1	26.2
TWIST [9]	5%	27.0	44.1	56.2
TWIST [9]	10%	30.6	49.7	63.0
TWIST [9]	20%	32.8	52.9	66.8
One Obj One Pt Supervision: SegGroup (PointGroup) [40]	0.02%	23.4	43.4	62.9
3D-WSIS [39]	0.02%	28.1	47.2	67.5
RWSeg (Ours)	0.02%	34.7	56.4	71.2

Table 3. Instance segmentation results on ScanNet-v2 [10] validation set. Methods marked with brackets represents using generated pseudo labels to train another fully-supervised method.

Method	Supervision	AP	AP ₅₀	AP ₂₅
Full Supervision: SoftGroup [43]	100%	50.4	76.1	86.5
H AIS [7]	100%	45.7	69.9	80.3
SSTNet [30]	100%	50.6	69.8	78.9
OccuSeg [19]	100%	48.6	67.2	78.8
PointGroup [25]	100%	40.7	63.6	77.8
3D-MPA [14]	100%	35.5	61.1	73.7
MTML [26]	100%	28.2	54.9	73.1
3D-BoNet [51]	100%	25.3	48.8	68.7
3D-SIS [24]	100%	16.1	38.2	55.8
GSPN [52]	100%	15.8	30.6	54.4
One Obj One Pt Supervision: SegGroup (PointGroup) [40]	0.02%	24.6	44.5	63.7
3D-WSIS [39]	0.02%	25.1	47.0	67.8
RWSeg (Ours)	0.02%	34.8	56.7	73.9

Table 4. Instance segmentation results on ScanNet-v2 [10] test set.

Method	Supv.	Area 5		6-fold	
		mPre	mRec	mPre	mRec
Full Supervision: PointGroup [25]	100%	61.9	62.1	69.6	69.2
One Obj One Pt Supervision: SegGroup (PointGroup) [40]	0.02%	47.2	34.9	56.7	43.3
3D-WSIS [39]	0.02%	50.8	38.9	59.3	46.7
RWSeg (Ours)	0.02%	60.1	45.8	68.9	56

Table 5. Instance segmentation results on S3DIS [4] dataset.

4.1. Ablation Study

In this section, we proceed to study the impacts of different components of our proposed method. Table 6 shows the network performance at different stages of training. We use “Self-Attn” to represent the self-attention module in our network. In the setting of “3D U-Net + Self-Attn”, we freeze the backbone network and only train self-attention module, which shows the effectiveness of this component. Stage 1 training is supervised by initial weak labels. And Stage 2 training is supervised by the generated pseudo labels from our algorithm at the end of Stage 1. With our training strategy, the quality of semantic features can be steadily improved.

mIoU	Method	train set	val set
Stage 1	3D U-Net	74.6	61.7
Stage 1	3D U-Net + Self-Attn	78.9	66
Stage 2	3D U-Net	80	68.4
Stage 2	3D U-Net + Self-Attn	81.6	70.3

Table 6. Ablation study for network components. “3D U-Net” indicates our backbone network, and “Self-Attn” means our proposed self-attention module for feature propagation. Evaluated on ScanNet-v2 [10] validation set.

Ablations on Cross-graph Competing Random Walks (CRW) To make fair comparisons on clustering algorithms for pseudo label generation, we train a PointGroup [25] backbone network with initial weak labels. On top of the shared network, we evaluate the performance of our CRW and other baseline methods in Table 7. “PointGroup BFS” represents a popular Breadth-First Search algorithm used in fully supervised 3D instance segmentation. K-means [20] is a simple yet powerful unsupervised clustering algorithm to separate samples in K groups of equal variance. Its character suits our task very well by nature. However, we found K-means is very sensitive to noise. The performance highly depends on the quality of semantic predictions and shift vectors. In contrast, our CRW is more robust and works well in different situations.

Figure 8 shows the change of seeding regions during the process of Cross-graph Competing Random Walks. At each step, the top 50% of the new predictions on unlabelled points are added as seed. It can be seen that new seeding

Baseline Methods	AP	AP ₅₀	AP ₂₅
PointGroup BFS [25]	15.8	32.4	58.9
K-means [†] [20]	14.5	28.5	66.9
K-means [‡] [20]	23.5	44.1	72.5
CRW[†] (Ours)	53.2	80.6	95.2
CRW[‡] (Ours)	55	82	95.9

Table 7. Comparison with pseudo label generation baseline methods on ScanNet-v2 [10] training set. Methods marked with [†] are based on original coordinates. Methods marked with [‡] are based on shifted coordinates. BFS uses both sets of coordinates.

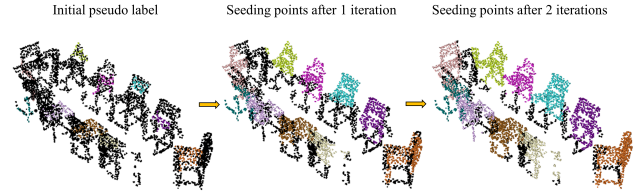


Figure 8. Visualized example of CRW’s seeding regions at different iterations.

points tend to be distributed at those regions relatively far from other seeds, as a result of cross-graph competition.

Iteration number (t_{2max})		0	1	5	10
chair	AP \uparrow	64.2	66.3	67.4	67.4
bookshelf	AP \uparrow	48.1	51.0	52.3	52.3

Table 8. Impact of the competing mechanism and iteration number on CRW ($\theta = 50\%$). Evaluated in AP for chair and bookshelf class on ScanNet-v2 [10] training set. $t_{2max} = 0$ represents the converged results from the basic RWSeg without competing mechanism.

The impact of using multiple random walk steps in Cross-graph Competing Random Walks (CRW) is shown in Table 8. As we expected, cross-graph competition is useful to resolve those ambiguous cases in instance segmentation, where objects from same category are compactly placed. Meanwhile, for those sparsely placed object categories, such as bathtub and door, their instance segments can already be well defined by proposed basic random walk algorithm. Competitions usually not exist for such cases.

5. Conclusion

In this paper, we propose a novel weakly supervised method for 3D instance segmentation on point clouds. With significantly fewer annotations, our network uses a self-attention module to propagate semantic features and a random walk based algorithm with cross-graph competition to generate high-quality pseudo labels. Comprehensive experiments show that our method achieves solid improvements on performance. The limitations of our method are discussed in the supplementary material.

Acknowledgement

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). This research work is also supported by the Agency for Science, Technology and Research (A*STAR) under its MTC Young Individual Research Grant (Grant No. M21K3c0130) and MTC Programmatic Funds (Grant No. M23L7b0021). This research is also partly supported by the MoE AcRF Tier 2 grant (MOE-T2EP20220-0007) and the MoE AcRF Tier 1 grant (RG14/22).

References

- [1] Point cloud instance segmentation with semi-supervised bounding-box mining. *CoRR*, abs/2111.15210, 2021. 1, 3, 7
- [2] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, pages 2209–2218, 2019. 3
- [3] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, pages 4981–4990, 2018. 3
- [4] 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 *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 6, 7, 8
- [5] Aditya Arun, CV Jawahar, and M Pawan Kumar. Weakly supervised instance segmentation by learning annotation consistent instances. In *ECCV*, pages 254–270, 2020. 3
- [6] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15467–15476, October 2021. 3
- [7] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *ICCV*, 2021. 7
- [8] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. Box2mask: Weakly supervised 3d semantic instance segmentation using bounding boxes. In *European Conference on Computer Vision (ECCV)*. Springer, October 2022. 1, 3, 7
- [9] Ruihang Chu, Xiaoqing Ye, Zhengzhe Liu, Xiao Tan, Xiaojuan Qi, Chi-Wing Fu, and Jiaya Jia. Twist: Two-way inter-label self-training for semi-supervised 3d instance segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1090–1099, 2022. 7
- [10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017. 1, 6, 7, 8
- [11] Shichao Dong and Guosheng Lin. Weakly supervised 3d instance segmentation without instance-level annotations, 2023. 3
- [12] Shichao Dong, Guosheng Lin, and Tzu-Yi Hung. Learning regional purity for instance segmentation on 3d point clouds. In *European Conference on Computer Vision*, pages 56–72. Springer, 2022. 3
- [13] Nico Engel, Vasileios Belagiannis, and Klaus Dietmayer. Point transformer. *IEEE Access*, 9:134826–134840, 2021. 3
- [14] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi proposal aggregation for 3d semantic instance segmentation, 2020. 3, 7
- [15] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 3
- [16] Zan Gojcic, Caifa Zhou, Jan D. Wegner, Leonidas J. Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3
- [17] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *CoRR*, abs/1706.01307, 2017. 2, 3, 6
- [18] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Apr 2021. 3
- [19] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation, 2020. 3, 7
- [20] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979. 8
- [21] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021. 3, 7
- [22] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, pages 7014–7023, 2018. 3
- [23] Maximilian Jaritz, Jia-Yuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. *ArXiv*, abs/1909.13603, 2019. 3
- [24] Hou Ji, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019. 3, 7
- [25] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation, 2020. 3, 4, 6, 7, 8
- [26] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3d instance segmentation via multi-task metric learning, 2019. 3, 7
- [27] Lei Li, Siyu Zhu, Hongbo Fu, Ping Tan, and Chiew-Lan Tai. End-to-end learning local multi-view descriptors for 3d point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

- [28] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, pages 820–830. Curran Associates, Inc., 2018. 3
- [29] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2783–2792, October 2021. 3
- [30] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2783–2792, 2021. 7
- [31] Yangbin Lin, Cheng Wang, Dawei Zhai, Wei Li, and Jonathan Li. Toward better boundary preserved supervoxel segmentation for 3d point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143:39–47, 2018. ISPRS Journal of Photogrammetry and Remote Sensing Theme Issue “Point Cloud Processing”. 4
- [32] Chen Liu and Yasutaka Furukawa. MASC: multi-scale affinity with sparse convolution for 3d instance segmentation. *CoRR*, 2019. 3
- [33] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1726–1736, June 2021. 1, 3
- [34] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, pages 1742–1750, 2015. 3
- [35] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [36] Pedro O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, pages 1713–1721, 2015. 3
- [37] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2016. 3
- [38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 3
- [39] Linghua Tang, Le Hui, and Jin Xie. Learning inter-superpoint affinity for weakly supervised 3d instance segmentation. In *ACCV*, 2022. 1, 2, 3, 7, 8
- [40] An Tao, Yueqi Duan, Yi Wei, Jiwen Lu, and Jie Zhou. Seg-Group: Seg-level supervision for 3D instance and semantic segmentation. *arXiv preprint*, 2020. 1, 2, 3, 7, 8
- [41] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *ArXiv*, abs/1904.08889, 2019. 3
- [42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 4
- [43] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. Softgroup for 3d instance segmentation on 3d point clouds. In *CVPR*, 2022. 7
- [44] 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. 3
- [45] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, 2018. 3
- [46] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, 2019. 3
- [47] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1
- [48] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4384–4393, 2020. 3, 7
- [49] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. *arXiv preprint arXiv:1811.07246*, 2018. 3
- [50] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, pages 13706–13715, 2020. 3
- [51] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds, 2019. 3, 7
- [52] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. *arXiv preprint arXiv:1812.03320*, 2018. 7
- [53] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, 2019. 3
- [54] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 4
- [55] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *CVPR*, pages 2209–2218, 2018. 3

Collaborative Propagation on Multiple Instance Graphs for 3D Instance Segmentation with Single-point Supervision (Supplementary Material)

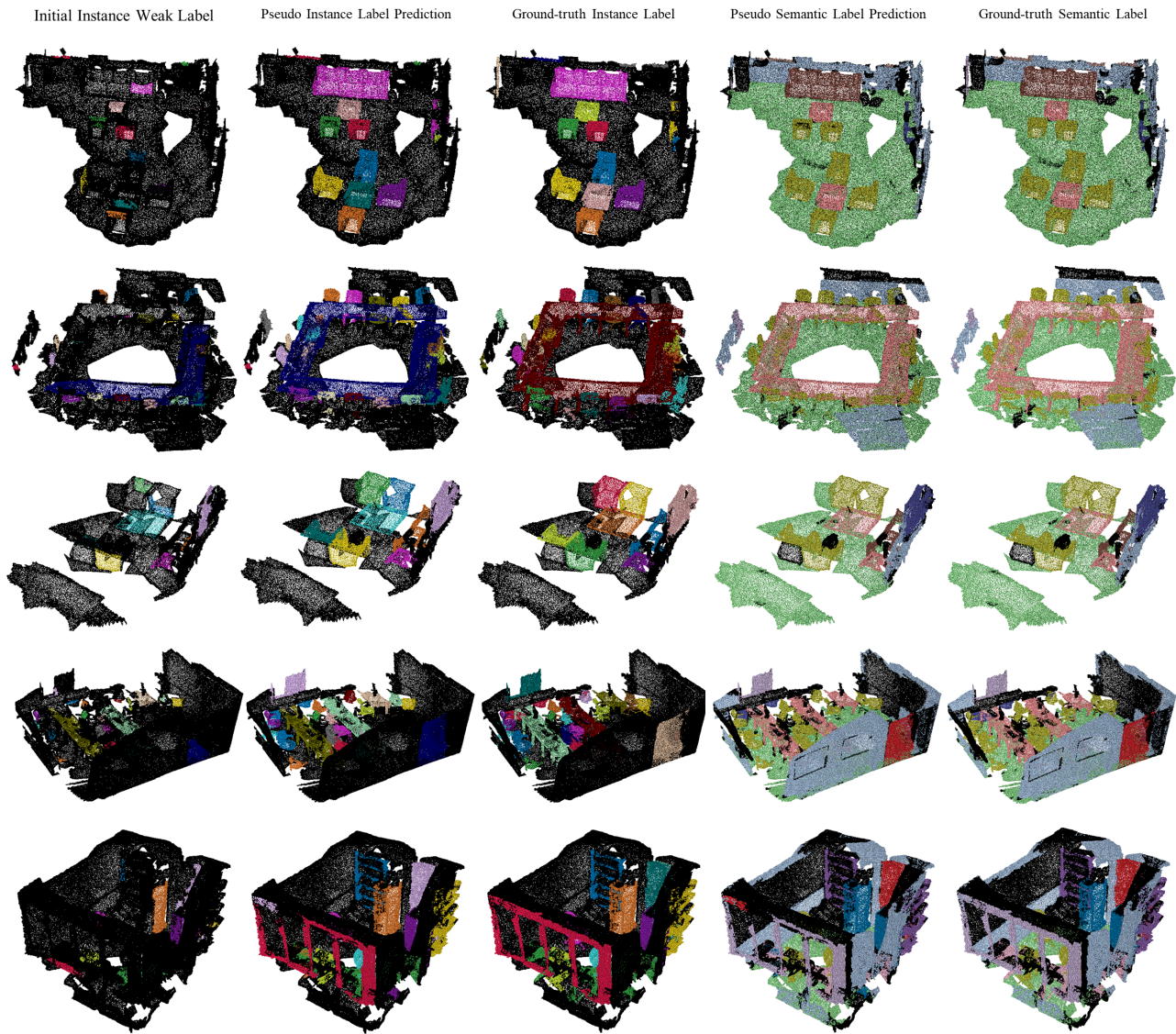


Figure 1. More Qualitative Comparison on ScanNet v2 [1] validation set.

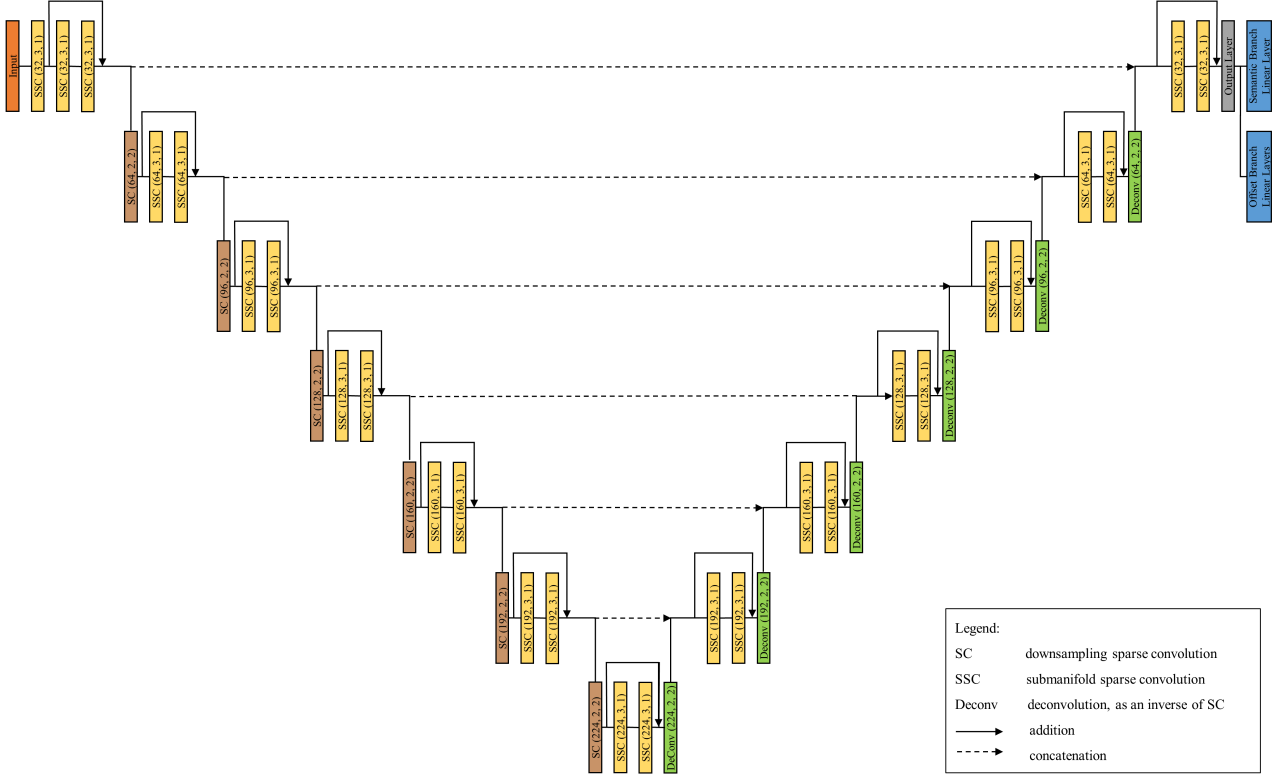


Figure 2. The detailed structure of 3D U-Net backbone with submanifold sparse convolution [2].

1. Additional Qualitative Results

In this section, we show some more visualization results of generated pseudo labels on the training set of ScanNet dataset [1]. As shown in Figure 1, initial instance weak labels are first derived from “one object one point” weak annotations. Then, the proposed method RWSeg can propagate information to unlabelled points. Generated pseudo labels are compared with fully annotated ground-truth for semantic segmentation and instance segmentation respectively. The results show our high-quality pseudo labels have very similar patterns to the actual annotations and contain only minor errors.

2. Network Architecture Details

In this section, we present the detailed structure of our 3D U-Net backbone with submanifold sparse convolution [2] and self-attention module. The backbone network is originally introduced by Graham [3] and has been widely used for feature extraction in point cloud segmentation tasks [4, 5, 6, 7, 8, 9]. The core idea of submanifold sparse convolution is to efficiently process spatially-sparse data, otherwise using normal 3D convolution can be very computationally expensive.

Backbone network In Figure 2, the backbone network takes the sparse voxelized representation of point cloud as input. The U-Net structure is mainly built based on sparse convolution (SC) layers and submanifold sparse convolution (SSC) layers. $SC(m, f, s)$ represents a downsampling sparse convolution (SC) layer with feature dimension m , kernel size f and stride s . Residual connection is used to contain two submanifold sparse convolution (SSC) layers. Deconvolution represents an inverse operation of sparse convolution (SC). The output of the backbone network is split into the semantic branch and offset branch. The semantic branch further utilizes a self-attention layer for feature propagation. For offset branch, point feature vectors are transformed via a two-layer MLP to the dimension of 3, which is then supervised by a regression loss for predicting the centroid shift vectors.

Self-attention module Figure 3 illustrates the process of representing each supervoxel set $\mathcal{V} = p_1, p_2, \dots, p_i$ as a super-point. This is achieved by performing an average pooling operation on both the semantic features S and the point coordinates

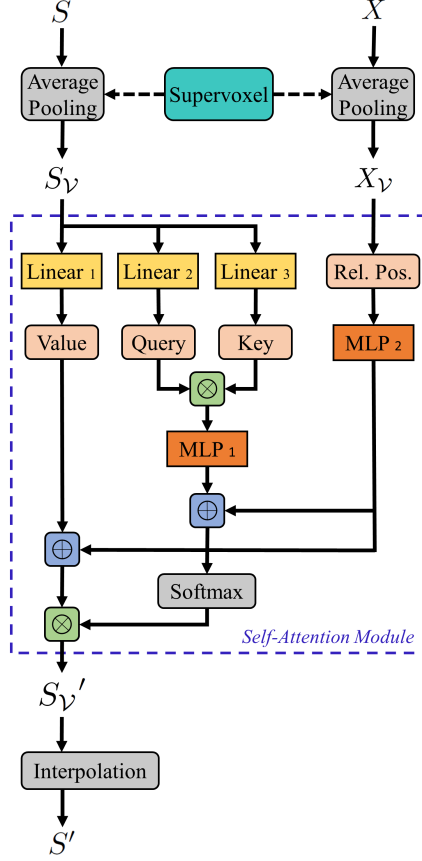


Figure 3. Illustration of self-attention module in semantic branch for feature propagation. \oplus denotes the broadcasting addition and \otimes denotes the element-wise multiplication. Rel. Pos. represents the relative positional similarity of input coordinates.

X for all points belonging to the set. Following [10, 11], we first perform linear transformations of the input semantic features S_V to three matrices as query, key, and value (Q, K, V). Then, matrix A captures the similarity between queries and keys and also includes encoded positional information for adjustment. This can be written as

$$\mathbf{Q} = S_V W_Q, \quad \mathbf{K} = S_V W_K, \quad \mathbf{V} = S_V W_V, \quad (1)$$

$$\mathbf{A} = \gamma\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) + \delta(X_V, X_V), \quad (2)$$

where d is the dimension of Q and V , X_V the coordinates of supervoxels, γ is a mapping function via MLP, δ is a positional similarity function via MLP. The output of self-attention can be formulated as

$$\text{Attn}(S_V) = \sigma(\mathbf{A})(\bar{\mathbf{V}} + \delta(X_V, X_V)), \quad (3)$$

where $\sigma(\cdot)$ is a softmax activation function. $\bar{\mathbf{V}}$ denotes a symmetric matrix created by repeatedly expanding \mathbf{V} .

Lastly, refined semantic features are interpolated to the original size in point cloud. The training process is supervised by a conventional cross-entropy loss H_{CE} with incomplete labels. We define the semantic loss as

$$L_{sem} = -\frac{1}{N} \sum_{i=1}^N H_{CE}(y_i, \hat{c}_i). \quad (4)$$

where \hat{c}_i is the weak semantic label. Unlabelled points are ignored here.

Offset loss function Following [6], We use a L_1 regression loss and a cosine similarity based direction loss to train the offset prediction,

$$L_{offset} = \frac{1}{\sum_i m_i} \sum_i \|d_i - (\hat{q}_i - p_i)\| \cdot m_i - \frac{1}{\sum_i m_i} \sum_i \frac{d_i}{\|d_i\|_2} \cdot \frac{\hat{q}_i - p_i}{\|\hat{q}_i - p_i\|_2} \cdot m_i. \quad (5)$$

where $\mathbf{m} = \{m_1, \dots, m_N\}$ is a binary mask. The value of m_i indicates whether point i is on an instance or not. This means we only consider foreground points with weak labels for supervision.

3. Ablations on Self-attention Module

In Table 1, we perform ablation study on self-attention module by blocking relative position feature on ScanNet v2 [1]. The structure with relative position feature broadcasting addition to both feature branch and attention branch can bring more performance gain.

Relative position usage	Train	Val
Baseline - backbone only	74.6	61.7
None	77.3	64.1
Feature branch only	77.6	64.3
Attention branch only	78.3	65.3
Feature branch + Attention branch	78.9	66

Table 1. Ablations on Self-attention Module

4. Random Walk with multiple Steps

This section explains how to inference the equation as the final steady-state of the random walk algorithm (From Eq.6 to Eq.7 in original paper).

Random walk algorithm is performed by repeatedly adjusting node vector b via transition matrix \mathbf{A} . At t -th iteration, the adjustment can be expressed as

$$\mathbf{b}_{t+1}^l = \alpha \mathbf{A} \mathbf{b}_t^l + (1 - \alpha) \mathbf{b}_0^l, \quad (6)$$

where b_t is the existing node vector derived at the previous random walk step, b_0 is the initial node vector, $\alpha \in [0, 1]$ is a blending coefficient between propagated score and initial score.

For random walk with multiple steps, we use t to represent the t -th iteration and Expand Eq. (6) to

$$\mathbf{b}_{t+1}^l = (\alpha \mathbf{A})^{t+1} \mathbf{b}_0^l + (1 - \alpha) \sum_{i=0}^t (\alpha \mathbf{A})^i \mathbf{b}_0^l. \quad (7)$$

Applying $t \rightarrow \infty$, since $\alpha \in [0, 1]$, the first term in Eq. (7) turns into

$$\lim_{t \rightarrow \infty} (\alpha \mathbf{A})^{t+1} \mathbf{b}_0^l = 0. \quad (8)$$

For the second term with matrices can be expanded as

$$\lim_{t \rightarrow \infty} \sum_{i=0}^t (\alpha \mathbf{A})^i \mathbf{b}_0^l = (\mathbf{I} - \alpha \mathbf{A})^{-1} \mathbf{b}_0^l, \quad (9)$$

where \mathbf{I} is the identity matrix. Thus, the final steady-state of random walk algorithm can be written as

$$\mathbf{b}_{(\infty)}^l = (1 - \alpha) (\mathbf{I} - \alpha \mathbf{A})^{-1} \mathbf{b}_0^l. \quad (10)$$

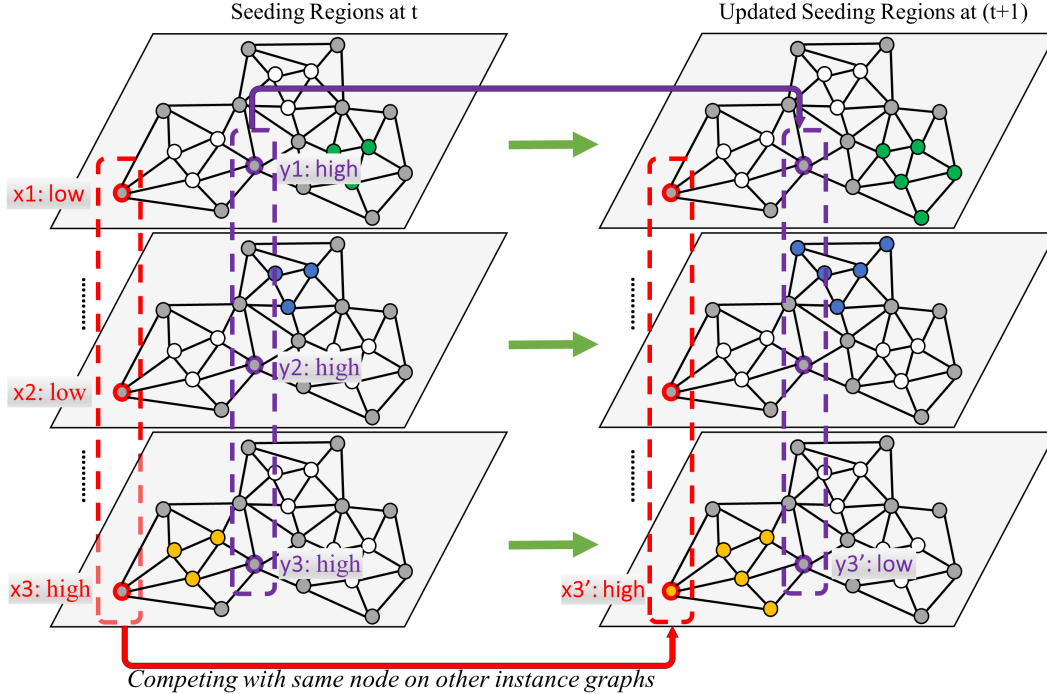


Figure 4. Illustration of competing mechanism in CRW. This example shows the effect of competition on two different nodes ($x3$ in red, $y3$ in purple).

5. Competing Mechanism in CRW

For illustrative purposes, we present an example in Figure 4. In this case, the foreground category consists of three instance graphs, each with a distinct seeding point marked in green, blue, and yellow, respectively. Node $x3$ (in red) has two nodes in the same position, $x1$ and $x2$. At step t , their node scores are determined by their overall distance to the seeding points. Since $x1$ is far from the seeding points marked in green, its score will be low after a random walk step, whereas $x3$, which is closer to the seeding points marked in yellow, will have a higher score. After applying SoftMax normalization to the scores of $x1$, $x2$, and $x3$, the output score for $x3'$ at step $t + 1$ will be high, as it faces less competition from the other two nodes.

Similarly, we have a node $y3$ with two same-positioned nodes, $y1$ and $y2$, placed in the center of three instances. At step t , the scores of $y1$, $y2$, and $y3$ are all high. However, during normalization, $y3$ receives a strong repulsive effect from $y1$ and $y2$. Thus, the output score $y3'$ at step $t + 1$ will be low.

Finally, the proposed algorithm compares the node scores at step $t + 1$. In this case, the node $x3'$ will have a higher priority to be grouped into seeds than $y3'$. This is because node $x3'$ is highly likely from the instance in yellow, whereas there is lower confidence in $y3'$. Therefore, we leave this node to be grouped in the later steps.

6. Ablations on hyperparameters in CRW

In Table 2, we show the experimental results with varying hyperparameters for the competing mechanism in CRW. The considered baseline is the proposed baseline random walk algorithm, which is represented by $t_{2max} = 0$.

The table illustrates that a lower update percentage θ typically leads to better results but requires more iterations t_{2max} , as it gradually groups the most confident points with our competing mechanism. The improvements over the random walk baseline are consistently observed. As discussed in the paper, the extent of the improvement depends on the distribution of the dataset. Notably, the proposed design in CRW is particularly effective in solving challenging cases, such as those with compacted objects of the same class.

Update percentage θ	Iteration number t_{2max}	AP (chair)	AP (bksf)
N.A.	0	64.2	48.1
80%	5	66.7 (+2.5)	49.9 (+1.8)
50%	5	67.4 (+3.2)	52.3 (+4.2)
20%	5	67 (+2.8)	53.4 (+5.3)
20%	20	67.3 (+3.1)	54.4 (+6.3)
10%	50	67.3 (+3.1)	55.1 (+7.0)

Table 2. Experiments with different CRW hyperparameters on ScanNet v2 [1]

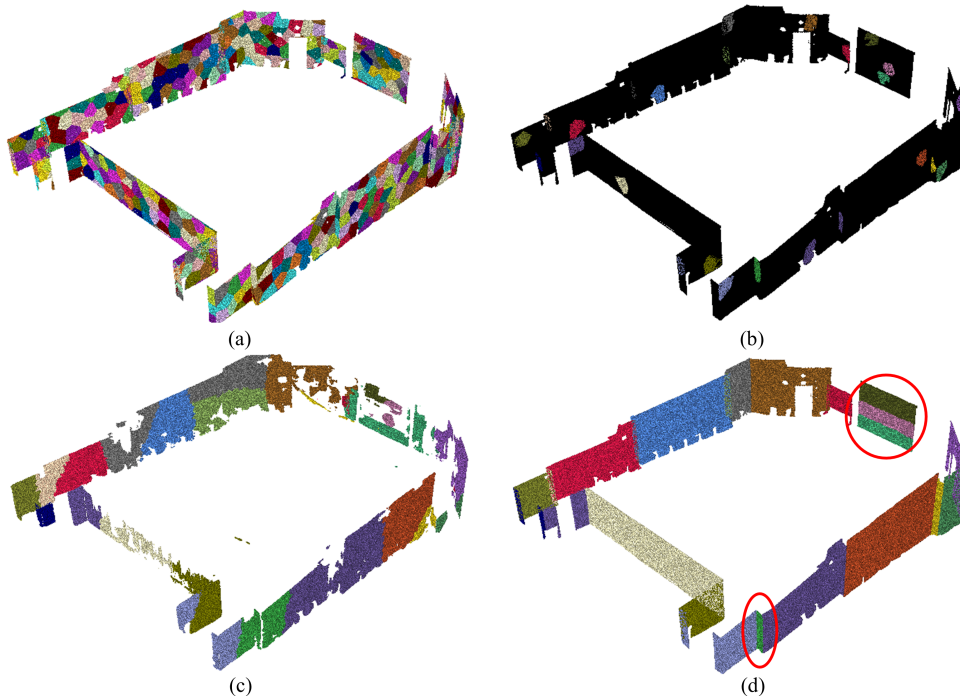


Figure 5. Limitations of RWSeg on S3DIS [12] dataset. (a) generated supervoxels (b) initial instance weak labels (c) generated instance pseudo labels (d) ground-truth instance labels

7. Additional Analysis in CRW

After conducting experiments with various values of hyperparameters for t_{1max} and α , we have observed that our algorithm can converge after just a single random walk step. Further increasing the iteration number of t_{1max} only results in marginal improvements. We argue that the fully connected graph used in our algorithm has a wide influence field. This means that it can exert its influence over a large area, and consequently, reduces the need for multiple random walk steps.

Hyperparameter $\alpha \in [0, 1]$ is used to control the trade-off between propagated node values and initial node values. Intuitively, it prevents deviating too fast from initial segmentation values. In our experiments, different values of α create a minor influence on the final converged results (less than 0.2% in mAP). However, if we set the value of α to 1 to remove the effect from initial values, the performance of random walk is dropped by 1% in mAP.

8. Limitations of RWSeg

In S3DIS [12] dataset, some background stuff such as walls, ceilings, boards are also treated as instances by their setting. As shown in Figure 5 (d), walls are intentionally labeled as separate instances, even though they are part of the background. Additionally, these walls can vary greatly in size, which poses a challenge for our method. Our method is primarily designed for common instance types and may struggle to make accurate predictions on these cases, especially with limited initial weak labels. In practice, one possible solution is to use surface normals as a clue and apply unsupervised plane estimation methods. However, this is beyond the scope of this work and goes beyond our objectives.

References

- [1] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 1, 2, 4, 6
- [2] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *CoRR*, abs/1706.01307, 2017. 2
- [3] Benjamin Graham, Martin Engelcke and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. *CoRR*, abs/1711.10275, 2017. 2
- [4] Chen Liu and Yasutaka Furukawa. MASC: multi-scale affinity with sparse convolution for 3d instance segmentation. *CoRR*, 2019. 2
- [5] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3d instance segmentation via multi-task metric learning, 2019. 2
- [6] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation, 2020. 2, 4
- [7] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation, 2020. 2
- [8] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15467–15476, October 2021. 2
- [9] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2783–2792, October 2021. 2
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3
- [11] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 3
- [12] 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 *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 6