One Thing One Click: A Self-Training Approach for Weakly Supervised 3D Semantic Segmentation

Zhengzhe Liu[1] Xiaojuan Qi[2] Chi-Wing Fu[1]
[1]The Chinese University of Hong Kong [2]The University of Hong Kong
{zzliu,cwfu}@cse.cuhk.edu.hk xjqi@eee.hku.edu.hk

Abstract

Point cloud semantic segmentation often requires large-scale annotated training data, but clearly, point-wise labels are too tedious to prepare. While some recent methods propose to train a 3D network with small percentages of point labels, we take the approach to an extreme and propose "One Thing One Click," meaning that the annotator only needs to label one point per object. To leverage these extremely sparse labels in network training, we design a novel self-training approach, in which we iteratively conduct the training and label propagation, facilitated by a graph propagation module. Also, we adopt a relation network to generate the per-category prototype and explicitly model the similarity among graph nodes to generate pseudo labels to guide the iterative training. Experimental results on both ScanNet-v2 and S3DIS show that our self-training approach, with extremely-sparse annotations, outperforms all existing weakly supervised methods for 3D semantic segmentation by a large margin, and our results are also comparable to those of the fully supervised counterparts.

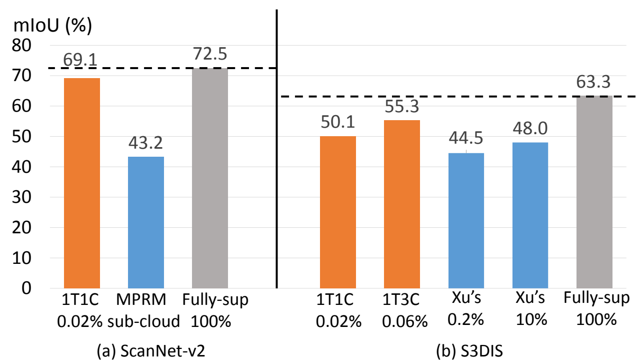

Figure 1. Comparing our approach of "One Thing One Click" (1T1C) with two recent weakly supervised methods MPRM [47] (CVPR 2020) and Xu's [50] (CVPR 2020) and a fully supervised version of our method Fully-sup on 3D semantic segmentation of ScanNet-v2 and S3DIS. Our approach achieves better performance by *training on data with only one label per object*. Note the annotation percentages under each method in the charts. If "One Thing Three Clicks" (1T3C) is allowed, we can further raise our result.

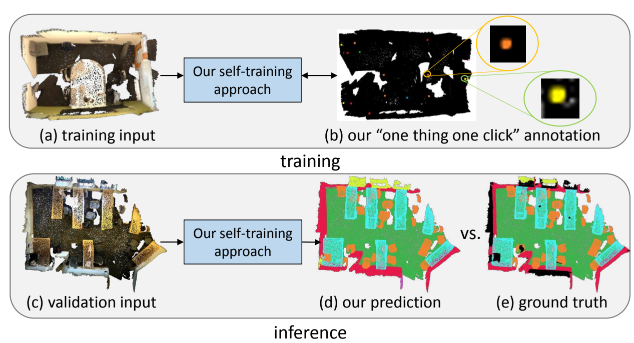

Figure 2. We train our self-training approach using only our "One Thing One Click" annotations (top). Yet, it can produce plausible segmentation results close to the ground truth (bottom).

1. Introduction

The success of 3D semantic segmentation benefits a lot from the large annotated training data. However, annotating a large amount of point cloud data is exhausting and costly. Taking ScanNet-v2[7] as an example, it takes 22.3 minutes to annotate one scene on average. It is a great burden to annotate the whole data set, which includes 1,513 scenes, thus potentially restricting further applications that require larger scale data. Thus, efficient approaches to facilitate 3D data annotation are highly desirable.

Very recently, some methods [47, 46, 50] were proposed to reduce efforts to annotating 3D point clouds. Though they improve annotation efficiency, various issues remain. Scene-level annotation in [47] could impose negative effects on the model in the absence of localization information, whereas sub-cloud annotation in [47] requires an extra burden to first divide the input into subclouds and then repeatedly annotate semantic categories in individual subclouds. The 2D image annotation approach [46] requires extra labor to prepare a 2D image annotation, which is also a tedious task on its own. Xu *et al.* [50] presume that the labeled points follow a uniform distribution. Such a requirement can be achieved by subsampling from a fully-annotated dataset, but is hard for the annotators to follow in practice.

In this work, we also aim to reduce the amount of necessary annotations on point clouds, but we take the approach to an extreme by proposing "One Thing One Click," so the annotator only needs to label one single point per object. To further relieve the annotation burden, such a point can be randomly chosen, not necessarily at the center of the object. On average, it takes less than 2 minutes to annotate a ScanNet-v2 scene with our "One Thing One Click" scheme (see an example annotation in Figure 2 (b), which contains only 13 clicks), which is more than 10x faster compared with the original ScanNet-v2 annotation scheme.

However, directly training a network on the extremely-sparse labels from our annotating scheme (less than 0.02% in ScanNet-v2 and S3DIS) will easily make the network overfit the limited data and restrict its generalization ability. Hence, it raises a question that "can we achieve a performance comparable with a fully supervised baseline given the extremely-sparse annotations?" To meet such a challenge, we design a self-training approach with a label-propagation mechanism for weakly supervised semantic segmentation. On the one hand, with the prediction result of the model, the pseudo labels can be expanded to unknown regions through our graph propagation module. On the other hand, with richer and higher quality labels being generated, the model performance can be further improved. Thus, we conduct the label propagation and network training iteratively, forming a closed loop to boost the performance of each other.

A core problem of label propagation is how to measure the similarity among nodes. Previous works [54, 5, 52] build a graph model upon 2D pixels and measure the similarity with low-level image features, e.g., coordinates and colors. In contrast, our graph is built upon the 3D super-voxels with more complex geometric structures and a variable number of points in each group. Hence, existing hand-craft features cannot fully reveal the similarity among nodes in our case. To resolve this problem, we further propose a relation network to leverage 3D geometrical information for similarity learning among the graph nodes in 3D. The geometrical similarity and learned similarity are integrated together to facilitate label propagation. To effectively train the relation network with the extremely-sparse and category-unbalanced data, we further propose to generate a category-wise prototype with a memory bank for better similarity measurement.

Experiments conducted on two public data sets ScanNet-v2 and S3DIS manifest the effectiveness of the proposed method. With just around 0.02% point annotations, our approach surpasses all existing weakly supervised approaches (which employ far more labels) for 3D point cloud segmentation by a large margin, and our approach even achieves results that are comparable with a fully supervised counterpart; see Figure 1. These results manifest the high efficiency of our "One Thing One Click" scheme for 3D point cloud annotation and the effectiveness of our self-training approach for weakly supervised 3D semantic segmentation.

2. Related Work

Semantic Segmentation for Point Cloud Approaches for 3D semantic segmentation can be roughly divided into point-based methods and voxel-based methods. *Point-based networks* take raw point clouds as input. Along this line of works, PointNet [33] and PointNet++ [34] are the pioneering ones. Afterward, convolution-based methods [23, 44, 48, 4] were also proposed for 3D semantic segmentation on point clouds. Recently, Kundu *et al*. [19] proposed to fuse features from multiple 2D views for 3D semantic segmentation. To aggregate together the geometrically-homogeneous points, Landrieu*et al*. [21] modeled a point cloud as a super point graph. Inspired by [21], we expand the sparse labels to geometrically homogeneous super-voxels to generate initial pseudo labels for the first-iteration network training.

Voxel-based networks take the regular voxel-grids as input instead of the raw data [43, 37, 11, 40, 8]. The recently-proposed methods SparseConv [12], MinkowskiNet *et al*. [6], and OccuSeg *et al*. [14] are among the representative works in this branch. In this paper, we adopt the 3D-UNet architecture described in [12] as the backbone architecture due to its high performance and applicability.

Weakly Supervised 3D Semantic Segmentation Compared with fully supervised 3D semantic segmentation, weakly supervised 3D semantic segmentation is relatively under-explored. After early works [28, 13] in this area, very recently, Wei *et al*. [47] utilized the Class Activation Map to generate pseudo point-wise labels from sub-cloud-level annotations. The performance is, however, limited by the lack of localization information. Wang *et al*. [46] back-projected 2D image annotations to 3D space to produce labels in point clouds. However, annotating large-scale semantic segmentation on 2D images is also laborious. Also, the visibility prediction branch adds to the complexity of the network. Xu *et al*. [50] achieve a performance close to fully supervised with less than 10% labels. However, they require the annotations to be uniformly-distributed in the point cloud, which is practically very hard for the annotators to follow.

Different from the existing works, we propose a new self-training approach with a graph propagation module, in which the network training and label propagation are conducted iteratively. Our approach largely reduces the reliance on the quality of the initial annotation and achieves top performances, compared with existing weakly supervised methods, while using only extremely-sparse annotations.

Self-Training Self-training for weakly supervised 2D image understanding has been intensively explored. To reduce

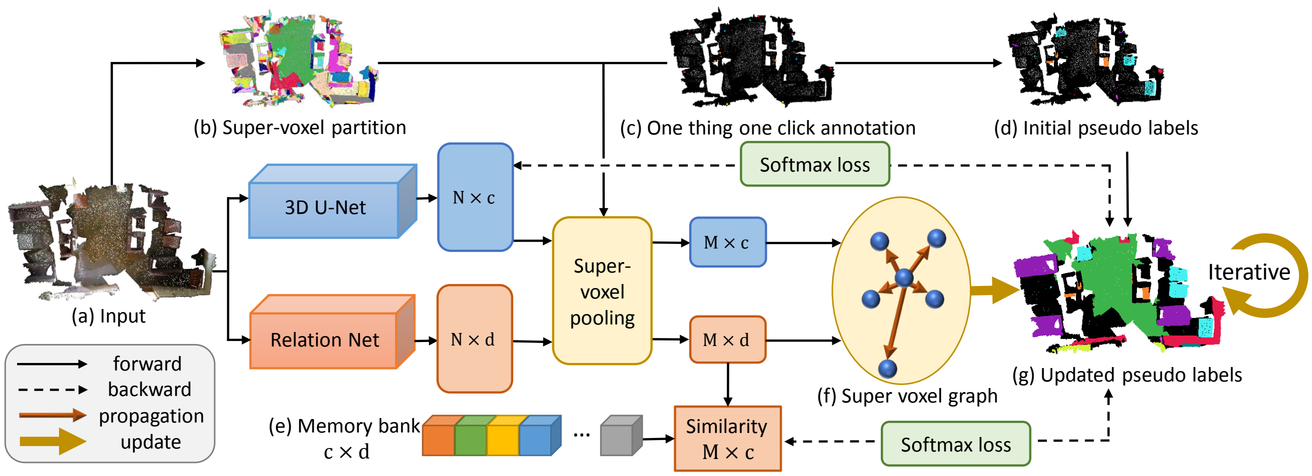

Figure 3. Overview of our framework. Through a super-voxel partition (b), we expand our "One Thing One Click" annotations (c) to generate the initial pseudo labels (d) for guiding the update of the pseudo labels (g). On the other hand, we adopt the "3D U-Net" for semantic label prediction (blue region) and design the "Relation Net" for super-voxel-based similarity learning (orange region). Then, we incorporate a super-voxel pooling to aggregate features from the two networks and construct the super-voxel graph (f) to propagate labels over the point cloud. Further, we iteratively update the predicted labels (g) and train the network through the softmax loss and contractive loss. C is the number of categories, D is the number of the feature dimension, N is the number of points, and M is the number of super-voxels.

the annotation burden for 2D images, researchers proposed a variety of annotation approaches, *e.g.*, image-level categories [35, 29, 55, 1], points [3, 22], extreme points [27, 31], scribbles [24, 45, 53], bounding boxes [9], etc. With the weak supervision, a self-training approach can learn to expand the limited annotations to unknown regions in the domain. As far as we know, this is the first work that explores self training for weakly supervised 3D semantic segmentation.

3. Methodology

3.1. Overview

With "One Thing One Click," we only need to annotate a point cloud with one point per object, as Figure 3 (c) shows, and these points can be chosen at random to alleviate the annotation burden. Procedure-wise, given such sparse annotations, we first over-segment the point cloud $X = \{p_i\}$ into geometrically homogeneous super-voxels $V = \{v_j\}$, where $\cup_j v_j = X$ and $v_j \cap v_{j'} = \emptyset$ for $v_j \neq v_{j'}$. Note that throughout the paper, we use i and j as the indices for points and super-voxels, respectively. Based on the super-voxel partition, we can produce initial pseudo labels of the point cloud by spreading each label to all the points locally in the super-voxel that contains the annotated point. However, as Figure 3 (d) shows, the labels are still very sparse. More importantly, the propagated labels distribute mainly around the initially-annotated points, which are far from the ideal uniform distribution for weakly semantic segmentation, as employed in [50].

An important insight in our approach is to iteratively propagate the sparse annotations to unknown regions in the point cloud, while training the network model to guide the propagation process. To achieve this, we adopt the 3D semantic segmentation network Θ (the blue regions in Figure 3) to learn to propagate via a graph model (Figure 3 (f)). Further, we design the relation network $\mathcal{R}$ (the orange regions in Figure 3) to explicitly model the feature similarity among the graph nodes. Afterward, predictions with high confidence are further employed as the updated pseudo labels for training the network in the next iteration (Figure 3 (g)). This iterative self-training approach couples the label propagation and network training, enabling us to significantly enhance the segmentation quality, as revealed earlier in Figure 1.

In this section, we first present our 3D semantic segmentation network for point-wise semantic prediction (Section 3.2), then our label propagation mechanism with a graph model and the relation network for similarity learning (Section 3.3). Afterward, we describe the self-training approach that evolves the above modules alternatively (Section 3.4).

3.2. 3D Semantic Segmentation Network

We adopt the 3D U-Net architecture [12] as the backbone, denoted as Θ. Its input is point cloud X of N points (Figure 3 (a)). Each point has 3D coordinates p_i and color c_i, where $i \in \{1, ..., N\}$. The network predicts the probability of each semantic category $P(y_{i,\bar{c}}|p_i, c_i, \Theta)$ of each point p_i, where $\bar{c}$ is the ground truth category of point p_i. The network is trained with the softmax cross-entropy loss below:

$$L_s = \frac{1}{N}\sum_{i=1}^{N} -\log P(y_{i,\bar{c}}|p_i, c_i, \Theta). \qquad (1)$$

In the first iteration, the network is trained with the initial pseudo labels, as shown in Figure 3 (d). In subsequent iterations, the network is trained with the updated pseudo labels, as shown in Figure 3 (g), which will be detailed below.

3.3. Pseudo Label Generation by Graph Propagation

To facilitate the network training, we propose a graph propagation mechanism to effectively propagate labels to unknown regions. We also propose the relation network to explicitly learn the similarity among the super-voxels to facilitate the label propagation process and complement 3D U-Net.

Graph Construction To start, we leverage the 3D geometrically homogeneous super-voxels to build a graph. Compared with building on points, our graph has significant fewer nodes to facilitate efficient label propagation.

To derive the prediction $P(y_{j,c}|v_j,\Theta)$ of the j-th super-voxel, we apply a super-voxel pooling to aggregate the semantic prediction of the n_j points in v_j as below:

$$P(y_{j,c}|v_j,\Theta) = \frac{1}{n_j}\sum_i P(y_{i,c}|p_i,c_i,\Theta),\ \text{where } p_i \in v_j, \tag{2}$$

where $P(y_{i,c}|p_i,c_i,\Theta)$ is the probability of p_i in class c.

To build the graph, we treat each super-voxel as a graph node and compute the similarity between each pair of super-voxels $v_j, v_{j'}$, which is represented as an edge. Further, to propagate labels to unknown regions through the graph, we formulate it as an optimization problem that considers both the network prediction and similarities among the super-voxels to achieve the global optimum with the energy function below similar to Conditional Random Field (CRF).

$$E(Y|V) = \sum_j \psi_u(y_j|V,\Theta) + \sum_{j<j'} \psi_p(y_j,y_{j'}|V,\mathcal{R},\Theta) \tag{3}$$

where $\mathcal{R}$ is the relation network to be detailed later. The unary term $\psi_u(y_j|V,\Theta)$ represents the super-voxel pooled prediction of the 3D U-Net $P(y_j)$ on super-voxel v_j. Specifically, it denotes the minus $\log$ probability of predicting super-voxel v_j to have label y_j. We define it as below.

$$\psi_u(y_j|V,\Theta) = -\log P(y_j|V,\Theta) \tag{4}$$

The pairwise term $\psi_p(j_k)$ in Equation 3 represents the similarity between super-voxels v_j and $v_{j'}$. We employ both the low-level features and learned features for measuring the similarity, as shown in Equation 5 below:

$$\psi_p(y_j,y_{j'}|V) = \mathbb{1}(y_j,y_{j'})\exp\{-\lambda_c\frac{\|c_j-c_{j'}\|^2}{2\sigma_c^2} -\lambda_p\frac{\|p_j-p_{j'}\|^2}{2\sigma_p^2} - \lambda_u\frac{\|u_j-u_{j'}\|^2}{2\sigma_u^2} - \lambda_f\frac{\|f_j-f_{j'}\|^2}{2\sigma_f^2}\} \tag{5}$$

where $\mathbb{1}(y_j,y_{j'})$ is 1, if v_j and $v_{j'}$ have different predicted labels, and 0 otherwise. The pairwise term means that the cost will be higher if super-voxels with similar features are predicted to be different classes. Here, $c_j, c_{j'}$, $p_j, p_{j'}$ and $u_j, u_{j'}$ are the normalized mean color, mean coordinates and mean 3D U-Net feature, respectively, of super-voxels v_j and $v_{j'}$. Unlike existing works [54, 5, 52], which build the graph on 2D image pixels, we build our graph on 3D super-voxels, which have irregular and complex geometrical structures, as shown in the supplementary material. Therefore, hand-crafted features $p_j, p_{j'}$ and $c_j, c_{j'}$ have inferior capability to measure the similarity between super-voxels. To address this issue, we propose the *Relation Network* to better leverage the 3D geometrical information and explicitly learn the similarity among super-voxels.

Relation Network Existing works Co-Training [36] and Tri-net [10] showed that semi-supervised training benefits from having two complementary tasks or components. In our framework, we propose a relation net to complement the 3D U-Net. The relation network $\mathcal{R}$ shares the same backbone architecture as the 3D U-Net Θ except for removing the last category-wise prediction layer. It aims to predict a category-related embedding f_j for each super-voxel v_j as the similarity measurement. Similar to Equation 2, f_j is the per super-voxel pooled feature in $\mathcal{R}$. In other words, the relation network groups the embeddings of same category together, while pushing those of different categories apart. To this end, we propose to learn a prototypical embedding for each category, inspired by the Prototypical Network [39].

However, the per-category prototypes in [39] are fully determined by the sampled mini-batch, and may deviate from the actual categorical center. Consequently, they may not be stable and could keep changing during the training, thereby hard to converge. To assist the training of the relation network with sparse and unbalanced training data, we present a memory bank $K = \{k\}$ to generate one categorical prototype for each category, instead of simply regarding the average embedding as the prototype as in [39].

The embedding f_j generated by $\mathcal{R}$ serves as a "query," and we compare it with the corresponding "key" k_c in the memory bank with a dot product. The two modules are optimized simultaneously with contrastive learning [30] as below.

$$L_c = \frac{1}{M}\sum_j^M(-\log\frac{f_j\cdot k_{\bar{c}}/\tau}{\sum_c f_j\cdot k_c/\tau}), \tag{6}$$

where τ is a temperature hyperparameter [49] and $\bar{c}$ is the ground truth category of v_j. The contrastive learning is equivalent to a c-way softmax classification task.

Following [15], we update the key representations via a moving average with momentum as shown below

$$k_{\bar{c}} \leftarrow mk_{\bar{c}} + (1-m)f_j, \tag{7}$$

where m is a momentum coefficient to control the evolving speed of the memory bank. On the one hand, the representations in the memory bank are initialized with random vectors, and are updated during training to generate the prototype for each category. On the other hand, the embeddings generated from the relation network are grouped towards the prototype of its category. In this way, the relation network generates similar embeddings for the same category and distinct ones for different categories. The memory bank updates the prototypes in a category-balanced manner by randomly sampling the same number of training samples s per category in every forward pass.

Our relation net complements with 3D U-Net. It measures the relations between super-voxels using different training strategies and losses, while 3D U-Net aims to project the inputs into the latent feature space for category assignment. The prediction of relation network is further combined with the prediction of 3D U-Net by multiplying the predicted possibilities of each category to boost the performance. In addition, the relation net offers a stronger measurement of the pairwise term in CRF vs. handcrafted features like colors and also complements with the 3D U-Net features.

3.4. Self-Training

With the energy function in Equation 3, we propose a self-training approach to update networks Θ and $\mathcal{R}$, and also the pseudo labels Y iteratively, as Algorithm 1 outlines. The self-training is started by the "One Thing One Click" annotations and the pre-constructed super-voxel graph. In each iteration, we fix network parameters $\Theta, \mathcal{R}$ and update label Y, and vice versa. There are two steps in each iteration.

- With Θ and $\mathcal{R}$ fixed, the label propagation is conducted to minimize the energy function in Equation 3. Then, the predictions with high confidence are taken as the updated pseudo labels for training the two networks in the next iteration. The confidence of super-voxel v_j, denoted as C_j, is the average of the minus $\log$ probability of all n_j points in v_j after the label propagation:

$$C_j = -\frac{1}{n_j}\sum_{i}^{n_j} \log P(y_i|p_i, V, \Theta, \mathcal{R}, G), \text{ where } p_i \in v_j, \tag{8}$$

  where G denotes the graph propagation.

- With pseudo labels Y, Θ and $\mathcal{R}$ are optimized with softmax loss and contrastive loss, respectively.

Algorithm 1: Our self-training approach.

Input : "One Thing One Click" annotations $Y_0 = \{p_i\}$; super-voxel partition $V = \{v_j\}$;
Output : semantic prediction for all points Y;

```
Expand the annotated points p_i to the super-voxel v_j if p_i ∈ v_j;
repeat
    Train 3D U-Net Θ with pseudo labels Y_t;
    Train relation network R with pseudo labels Y_t;
    Combine the predictions and propagate the label with the graph model;
    Update the pseudo labels Y_t to Y_{t+1} with the predictions of high confidence.
until convergence;
```

4. Experiments

Datasets Our experiments are conducted on two large 3D semantic segmentation datasets – ScanNet-v2 [7] and S3DIS [2]. **ScanNet-v2** [7] contains 1513 3D scans of 20 semantic categories. We annotate the official training set with our "One Thing One Click" scheme, and evaluate on the original validation and test set. **S3DIS** [2] contains 3D scans of 271 rooms containing 13 categories. We follow the official train/validation split to annotate on Area 1,2,3,4,6 and report the performance on Area 5.

"One Thing One Click" Annotation Details In order to ensure the randomness of point selection in annotation, we simulate the annotation procedure by selecting a single point inside an object with the same probability for the following experiments. In ScanNet-v2, only 19.74 points per scene are annotated on average with "One Thing One Click" scheme, while this number in the original ScanNet-v2 is 108875.9. In S3DIS, only 36.15 points in each room are annotated on average using "One Thing One Click", while the original S3DIS has 193797.1 points annotated in each room.

Implementation Details We implement all the modules of our self-training framework including the mean-field solver [18] for label propagation with the PyTorch [32] framework based on the implementation of [17]. Following [17], due to the GPU capacity, we randomly choose 250k points if the scene contains more points in training. In inference, the network takes the whole scene as input. We use the mesh segment results [7] as super-voxels for ScanNet-v2, and the geometrical partition results described in [21] for S3DIS super-voxel partition. We set the hyper-parameters $D = 32$, $T = 0.9$, $s = 20$, $\tau = 0.07$, $m = 0.9$, $\sigma_c = \sigma_p = \sigma_u = \sigma_f = 1$, $\lambda_c = \lambda_p = \lambda_u = \lambda_f = 1$ with a small validation set. We found that the self-training converges after five iterations. After that, more iterations training only brings very minor improvements.

4.1. Evaluations on ScanNet-v2

Comparing with Existing Methods Table 1 reports the benchmark result on ScanNet-v2 test set. The baselines can be roughly divided into two branches. (i) Fully supervised approaches with 100% supervision, including several

Method	Supervision	mIoU (%)
Pointnet++ [34]	100%	33.9
SPLATNet [40]	100%	39.3
TangentConv [42]	100%	43.8
PointCNN [23]	100%	45.8
FPConv [25]	100%	63.9
DCM-Net [38]	100%	65.8
PointConv [48]	100%	66.6
KPConv [44]	100%	68.4
JSENet [16]	100%	69.9
SubSparseCNN [12]	100%	72.5
MinkowskiNet [6]	100%	73.6
Virtual MVFusion [19]	100%+2D	74.6
Our fully-sup baseline	100%	72.5
MPRM [47]	scene-level	24.4
MPRM [47]	subcloud-level	41.1
MPRM+CRF [47]	subcloud-level	43.2
One Thing One Click	0.02%	**69.1**
Ours on "Data Efficient"	20 points/scene	**59.4**

Table 1. Comparing with existing methods and baselines on ScanNet-v2 test set.

representative works in 3D semantic segmentation. These methods are the upper bounds of weakly supervised ones. (ii) Weakly supervised approaches, including a recent work [47].

With less than 0.02% annotated points, our result (69.1% mIoU) outperforms many existing works with full supervision. As for weakly supervised approaches, MPRM [47] is trained with scene-level or subcloud-level labels. The scene-level annotation leads to an inferior performance of 24.4%, and the subcloud-level annotation takes around 3 minutes per scene as reported in [47], which is longer than our "One Thing One Click" scheme (2 minutes). More importantly, our result outperforms [47] by more than 26% mIoU.

Comparing with Our Baselines In this section, we first present three important baselines as shown in Table 2 on ScanNet-v2 validation set.

- Table 2 "Our fully sup baseline" is trained with the official 100% annotation provided by ScanNet-v2. It serves as the upper bound of our method.
- The model directly trained with the raw annotated points as Figure 3 (c) cannot converge well due to the extreme sparsity of the training data.
- Table 2 "One Thing One Click*". The model trained with the initial pseudo labels as Figure 3 (d) achieves 62.18% mIoU. It serves as the starting point of our self-training approach and is denoted as "our baseline" in the following.

Table 2 "One Thing One Click" manifests that our self-training approach surpasses the baseline by nearly 10% mIoU, attaining a 16% relative improvement. Compared with the fully supervised baseline with the same network architecture, our performance is only 2% lower.

Setting	Annotation	mIoU (%)
Our fully sup baseline	100%	72.18
One Thing One Click*	0.02%	62.18
One Thing One Click†	0.02%	68.96
One Thing One Click	0.02%	**70.45**
Data Efficient*	20 points	55.06
Data Efficient†	20 points	59.98
Data Efficient	20 points	**61.35**

Table 2. Our results and baselines on ScanNet-v2 val. set. * means the baseline model trained with the initial pseudo labels shown in Figure 3 (d). † means disabling graph propagation and relation network during inference, but note that they are still used in training.

Table 2 "One Thing One Click†" refers to disabling the graph propagation and relation network in inference. Note that they are still being used in training for generating the pseudo labels. This brings no extra computational burden during the inference, but helps to improve nearly 7% mIoU, comparing with the baseline (68.96% vs 62.18%).

The quantitative results in Figure 5 indicate our result (c) is very similar to the fully supervised baseline (e) [12] in ScanNet-v2. Check error maps (d) (f) for better comparison.

Results on ScanNet-v2 Data-Efficient Benchmark In this section, we show results on ScanNet-v2 "3D Semantic label with Limited Annotations" benchmark. We report the results on the most challenging setting with only 20 points annotated each scene in Table 1 "Ours on Data Efficient" and Table 2 "Data Efficient". In this experiment, we use the officially provided 20 points instead of "One-Thing-One-Click", and then employ our self-training approach for semantic segmentation. Note that we are the first to report results on this benchmark. The results show that our approach is not limited to "One-Thing-One-Click" and is applicable to other annotation schemes. However, the performance is inferior to "One-Thing-One-Click", since the annotations are more uneven among the categories.

Ablation Studies To further study the effectiveness of self-training, graph propagation and relation network, we conduct ablation studies on these three modules on ScanNet-v2 validation set as shown in Table 3 with single view evaluation.

"3D U-Net" indicates that the labels are propagated only based on the confidence score of the 3D U-Net itself, *i.e.*, the unary term in Equation 3. This ablation is designed to manifest the effectiveness of self-training. The "3D U-Net" column in Table 3 manifests that the performance is consistently improved with self-training strategy even without pairwise energy term in Equation 3 and super-voxel partition.

"3D U-Net+GP" refers to the label propagation with

Figure 4. Pseudo labels for each iteration on ScanNet-v2 training set.

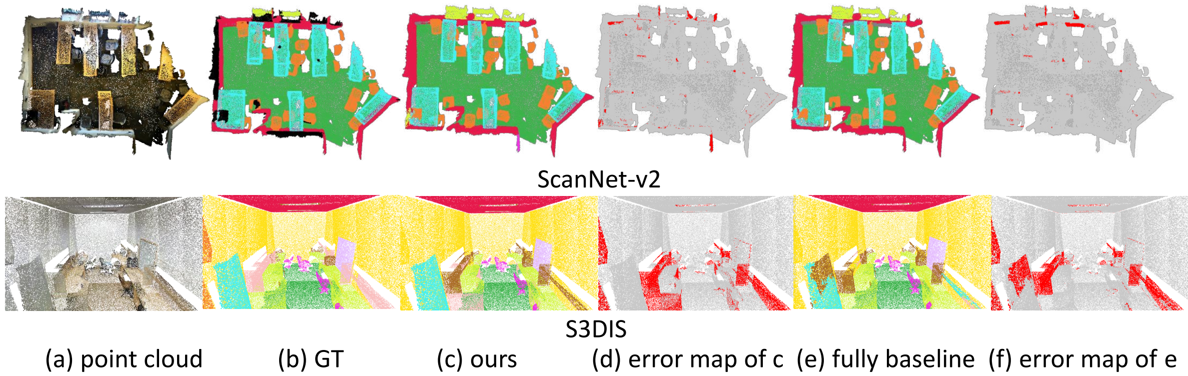

Figure 5. Quantitative results of our method and fully supervised baseline. (d) is the error map of our prediction (c), and (f) is the error map of our fully supervised baseline [12] (e). Red regions indicate the wrong prediction.

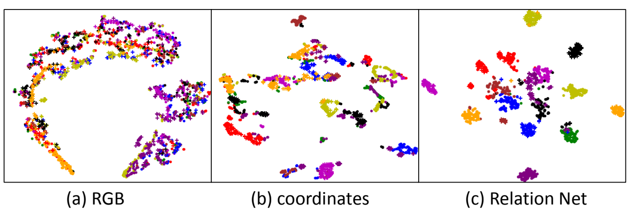

Figure 6. The t-SNE visualization of super-voxel features. Different colors and marks (point and plus) indicate different categories. The samples of the same category are better grouped together with our relation network (c), compared with hand-crafted features (a & b).

Method	3D U-Net	3D U-Net+GP	3D U-Net+Rel+GP
Iter1	60.14	63.83	**63.92**
Iter2	62.39	64.74	**66.97**
Iter3	64.83	66.10	**68.40**
Iter4	65.81	67.78	**70.01**
Iter5	65.91	67.92	**70.45**

Table 3. Ablation studies. "GP" indicates the graph propagation, and "Rel" means the relation network. "3D U-Net " refers to propagating labels only with the network prediction itself. "3D U-Net+GP" indicates label propagation with hand-crafted features. "3D U-Net+Rel+GP" indicates label propagation with our relation network. Evaluated on ScanNet-v2 val. set with single view testing.

graph model, and the similarity among super-voxels are measured by the coordinates p_i and colors c_i without the learned feature f_i. This ablation study is to show the effectiveness of the graph model. The results in Table 3 indicate that the graph model benefits the label propagation, and finally boosts the overall performance by 2% over "3D U-Net" (67.92% vs. 65.91%).

"3D U-Net+Rel+GP" utilizes the relation network for similarity measurement based on "3D U-Net+GP". In this setting, the similarity among super-voxels is measured with the averaged coordinates p_i, the colors c_i, the unary features u_i, and the relation network generated feature f_i, as shown in Equation 3. This experiment is to manifest that the relation network benefits the similarity measurement and pseudo label generation, compared with the hand-crafted feature, i.e., coordinates and color. It outperforms the hand-crafted features especially in the later iterations since the network benefits from the richer pseudo labels. It finally achieves 2.5% improvement compared with "3D U-Net+GP" (70.45% vs. 67.92%). As shown in Figure 4, the generated pseudo labels for each iteration expands to unknown regions step by step and finally gets close to the ground truth.

Analysis of Relation Network Further, we study whether the learned embeddings of the relation network outperform the hand-crafted features for similarity measurement. We randomly sample 200 super-voxels for each category in ScanNet-v2, and conduct a t-SNE visualization [26] on them. Figure 6 indicates that the relation network better groups the intra-class embeddings and distinguish the inter-class embeddings compared with hand-crafted features.

4.2. Evaluations on S3DIS

We also evaluate our annotation and training approach on the S3DIS dataset. Only less than 0.02% points in the dataset are annotated with our "One Thing One Click" scheme. To study whether the performance can be further boosted with richer annotations, we additionally conduct a "One Thing Three Clicks" scheme on S3DIS, where random 3 points per-object are annotated.

Comparing with Existing Works We also compare with fully supervised approaches and weakly supervised approaches on S3DIS. The latter includes existing works [20, 41] and recent works [50, 46].

As shown in Table 4, with the "One Thing One Click" scheme where less than 0.02% points are annotated, we achieve 50.1% mIoU. With "One Thing Three Clicks" scheme, our performance can be further improved to 55.3% mIoU. The above two results outperform [50] by 5.6% and 10.8% mIoU (0.2% annotations in [50]), and 2.1% and 7.3% mIoU (10% annotations in [50]) respectively.

Wang *et al*. [46] unprojects 2D semantic labels to 3D space for 3D semantic segmentation. To compare with [46], we first compare with the actual number of annotated points regardless of 2D or 3D. For S3DIS, the number of annotated 2D pixels (70,496 images with 1080×1080 resolution) is 100× more than the officially annotated 3D points (5.27×10^8 in total), so both settings of [46] (100% 2D annotations and 16.7% 2D annotations) actually utilize a large quantity of annotations. Even with a large gap of annotation, the results in Table 4 show that our "One Thing Three Clicks" scheme with only 0.06% 3D annotation outperforms [46] with 100% 2D annotations by nearly 3% mIoU.

In addition, our approach achieves comparable results with several fully supervised methods as shown in Table 4.

Comparing with Our Baselines We follow the similar settings in Section 4.1 to show several baselines for S3DIS.

- Table 4 "Our fully-sup baseline". The model trained with the full supervision of S3DIS achieves 63.7% mIoU. It serves as the upper bound of our approach.
- The model directly trained with only the annotated points in Figure 3 (c) cannot converge well.
- Table 4 "One Thing One Click*" and "One Thing Three Clicks*". The model trained with the annotated supervoxels in Figure 3 (d) achieves 43.7% mIoU for "One Thing One Click" and 48.9% mIoU for "One Thing Three Clicks". They are used as the baselines to calculate the "relative improvement" of our approach, and are denoted as "our baseline" in the following.

As shown in Table 4 "Rel. Imp." column, we have 14.6% ("One Thing One Click") and 13.1% ("One Thing Three Clicks") relative improvement over our baseline, surpassing the relative improvement of [50], which is 1.1% (with 0.2% annotations) and 5% (with 10% annotations) over their own baselines, by a large margin. The significant improvement of "relative improvement over baseline" manifests the effectiveness of the proposed approach.

Method	Supervision (%)	mIoU(%)	Rel. Imp. (%)
PointNet [33]	100%	41.1	-
SegCloud [43]	100%	48.9	-
TangentConv [42]	100%	52.8	-
3D RNN [51]	100%	53.4	-
PointCNN [23]	100%	57.3	-
SuperpointGraph [21]	100%	58.0	-
MinkowskiNet32 [6]	100%	65.4	-
Virtual MV-Fusion [19]	100%+2D	65.4	-
Our fully-sup baseline	100%	63.7	-
Π Model [20]	0.2%	44.3	-
MT [41]	0.2%	44.4	-
Xu *et al*. [50]*	0.2%	44.0	-
Xu *et al*. [50]	0.2%	44.5	1.1
Π Model [20]	10%	46.3	-
MT [41]	10%	47.9	-
Xu *et al*. [50]*	10%	45.7	-
Xu *et al*. [50]	10%	48.0	5.0
GPFN [46]	16.7% 2D	50.8	-
GPFN [46]	100% 2D	52.5	-
One Thing One Click*	0.02%	43.7	-
One Thing One Click†	0.02%	49.4	**13.0**
One Thing One Click	0.02%	**50.1**	**14.6**
One Thing Three Clicks*	0.06%	48.9	-
One Thing Three Click†	0.06%	54.1	10.6
One Thing Three Clicks	0.06%	**55.3**	13.1

Table 4. Comparison with existing methods and baselines on the S3DIS Area-5. * indicates baseline models, and † refers to disabling graph propagation and relation network during inference. Note that they are still used in training. "Rel. Imp." indicates the relative improvement over the baseline. "-" indicates there is no meaningful baseline in this case or it is a baseline itself.

To evaluate without any extra computation burden, we further disable the label propagation and relation network in inference as shown in Table 4 "†". Note that they are still adopted in training. Our model still attains 13.0% ("One Thing One Click") and 10.6% ("One Thing Three Clicks") relative improvement over our baseline in this case.

5. Conclusion

We propose the "One Thing One Click" scheme to efficiently annotate point clouds for weakly supervised 3D semantic segmentation, requiring significantly fewer annotations than the previous approaches. To put this scheme into practice, we formulate a self-training approach to make it feasible for the network to learn from such extremely sparse labels. Specifically, we execute the two key modules in our approach iteratively: expand labels through the graph propagation module and train the network using the updated pseudo labels. Further, we adopt a relation network to explicitly learn the feature similarity among graph nodes with complex 3D structures. Experiments on two large 3D datasets ScanNet-v2 and S3DIS manifest that our approach, with only extremely-sparse annotations, outperforms all the existing weakly supervised methods on 3D semantic segmentation by a large margin, and our results are also comparable to those of the fully supervised counterparts.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4981–4990, 2018.

[2] Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[3] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision (ECCV)*, pages 549–565. Springer, 2016.

[4] Alexandre Boulch. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 40(4):834–848, 2017.

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019.

[7] 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.

[8] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018.

[9] Jifeng Dai, Kaiming He, and Jian Sun. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 1635–1643, 2015.

[10] Wei Gao Dong-Dong Chen, Wei Wang and Zhi Hua Zhou. Tri-net for semi-supervised deep learning. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2018.

[11] Ben Graham. Sparse 3D convolutional neural networks. *arXiv preprint arXiv:1505.02890*, 2015.

[12] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] Stéphane Guinard and Loic Landrieu. Weakly supervised segmentation-aided classification of urban scenes from 3D lidar point clouds. In *ISPRS Workshop 2017*, 2017.

[14] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3D instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2940–2949, 2020.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020.

[16] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. JSENet: Joint semantic segmentation and edge detection network for 3D point clouds. *arXiv preprint arXiv:2007.06888*, 2020.

[17] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3D instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4867–4876, 2020.

[18] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[19] Abhijit Kundu, Xiaoqi Michael Yin, Alireza Fathi, David Alexander Ross, Brian Brewington, Tom Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3D semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[21] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, 2018.

[22] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 547–562, 2018.

[23] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *Advances in neural information processing systems (NeurIPS)*, pages 820–830, 2018.

[24] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3159–3167, 2016.

[25] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. FPConv: Learning local flattening for point convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4293–4302, 2020.

[26] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[27] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 616–625, 2018.

[28] Jilin Mei, Biao Gao, Donghao Xu, Wen Yao, Xijun Zhao, and Huijing Zhao. Semantic segmentation of 3D lidar data in dynamic scene using semi-supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2496–2509, 2019.

[29] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency for object segmentation from image level labels. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5038–5047. IEEE, 2017.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[31] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 4930–4939, 2017.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS) 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[33] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 652–660, 2017.

[34] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems (NeurIPS)*, pages 5099–5108, 2017.

[35] Xiaojuan Qi, Zhengzhe Liu, Jianping Shi, Hengshuang Zhao, and Jiaya Jia. Augmented feedback in semantic segmentation under image level supervision. In *European conference on computer vision (ECCV)*, pages 90–105. Springer, 2016.

[36] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–152, 2018.

[37] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3577–3586, 2017.

[38] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. DualConvMesh-Net: Joint geodesic and euclidean convolutions on 3D meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8612–8622, 2020.

[39] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems (NeurIPS)*, pages 4077–4087, 2017.

[40] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SplatNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2530–2539, 2018.

[41] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems (NeurIPS)*, pages 1195–1204, 2017.

[42] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2018.

[43] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3D point clouds. In *2017 international conference on 3D vision (3DV)*, pages 537–547. IEEE, 2017.

[44] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6411–6420, 2019.

[45] Bin Wang, Guojun Qi, Sheng Tang, Tianzhu Zhang, Yunchao Wei, Linghui Li, and Yongdong Zhang. Boundary perception guidance: A scribble-supervised semantic segmentation approach. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3663–3669, 2019.

[46] 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.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4384–4393, 2020.

[48] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, 2019.

[49] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, 2018.

[50] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13706–13715, 2020.

[51] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 403–417, 2018.

[52] Hao Yuan and Shuiwang Ji. StructPool: Structured graph pooling via conditional random fields. In *International Conference on Learning Representations (ICLR)*, 2019.

[53] Jing Zhang, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, and Yuchao Dai. Weakly-supervised salient object detection via scribble annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12546–12555, 2020.

[54] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 1529–1537, 2015.

[55] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3791–3800, 2018.