An MIL-Derived Transformer for Weakly Supervised Point Cloud Segmentation

Cheng-Kun Yang[1] Ji-Jia Wu[2] Kai-Syun Chen[2] Yung-Yu Chuang[1] Yen-Yu Lin[2,3]
[1]National Taiwan University [2]National Yang Ming Chiao Tung University [3]Academia Sinica

Abstract

We address weakly supervised point cloud segmentation by proposing a new model, MIL-derived transformer, to mine additional supervisory signals. First, the transformer model is derived based on multiple instance learning (MIL) to explore pair-wise cloud-level supervision, where two clouds of the same category yield a positive bag while two of different classes produce a negative bag. It leverages not only individual cloud annotations but also pair-wise cloud semantics for model optimization. Second, Adaptive global weighted pooling (AdaGWP) is integrated into our transformer model to replace max pooling and average pooling. It introduces learnable weights to re-scale logits in the class activation maps. It is more robust to noise while discovering more complete foreground points under weak supervision. Third, we perform point subsampling and enforce feature equivariance between the original and subsampled point clouds for regularization. The proposed method is end-to-end trainable and is general because it can work with different backbones with diverse types of weak supervision signals, including sparsely annotated points and cloud-level labels. The experiments show that it achieves state-of-the-art performance on the S3DIS and ScanNet benchmarks. The source code will be available at https://github.com/jimmy15923/wspss_mil_transformer.

1. Introduction

Point clouds capture geometric characteristics and surface context, and hence serve as an essential representation for many 3D vision applications such as scene understanding [6, 22, 28], autonomous vehicles [4, 5], and robotics [9]. Point cloud segmentation aims to identify points belonging to semantic categories of interest. It offers point-level recognition, thereby being an intrinsic component of point cloud analysis. However, learning a segmentation model relies on training data with point-level annotations. The high annotation cost poses an obstacle to this task. To address this issue, existing weakly supervised methods derive the segmentation model with different weak supervisory signals, such as partially labeled points [26, 42, 46, 47], sub-cloud level annotations [38] or scene level annotations [31].

To compensate for the lack of complete annotations, weakly supervised point cloud segmentation methods [26, 31, 38, 42, 46, 47] make the most of weakly labeled data by different techniques such as graph propagation, permutation consistency, and object proposals. Despite effectiveness, these methods use only *intra-cloud* information: The supervisory signals are grabbed from point clouds independently. Inspired by image co-segmentation [13,45] and cross-image pattern mining [32], we aim to explore *inter-cloud* semantics to supervise segmentation model training. To this end, we generalize the transformer model [34] to work on paired point clouds and formulate the problem as a *multiple instance learning (MIL)* [27] task. It follows that our method can use both intra-cloud and inter-cloud information to better accomplish weakly supervised segmentation.

Specifically, we develop an MIL-derived transformer where MIL addresses the uncertainty of weak labels. As shown in Figure 1, we apply the transformer to two point clouds of the same category. One cloud is treated as an anchor with each of its points being a *query* in the transformer. The other cloud serves as a reference where each of its points forms a *key-value* pair. The transformer encoder and decoder are applied to the reference and the anchor respectively. Through the cross-attention [34] mechanism of the decoder, each query (from the anchor) is expressed as a weighted sum of the values (from the reference). The resultant feature vectors of all points (*i.e.*, queries) of the anchor yield a *positive bag* in MIL for the common category of the two clouds. MIL via this positive bag encourages the model to attend to the foreground points of the anchor.

In contrast, we consider another reference point cloud whose category is different from the anchor. This time, the feature vectors of all queries of the anchor generate a *negative bag* because the feature vector of each query is a weighted sum of values of this reference where the target category is not present. The model via this negative bag is enforced to suppress irrelevant points. With the proposed MIL-transformer, each pair of point clouds, forming either a positive or a negative bag, produces extra signal to supervise model training. In addition, long-range dependency can be taken into account via the transformer.

Max or average pooling is widely used to aggregate in-

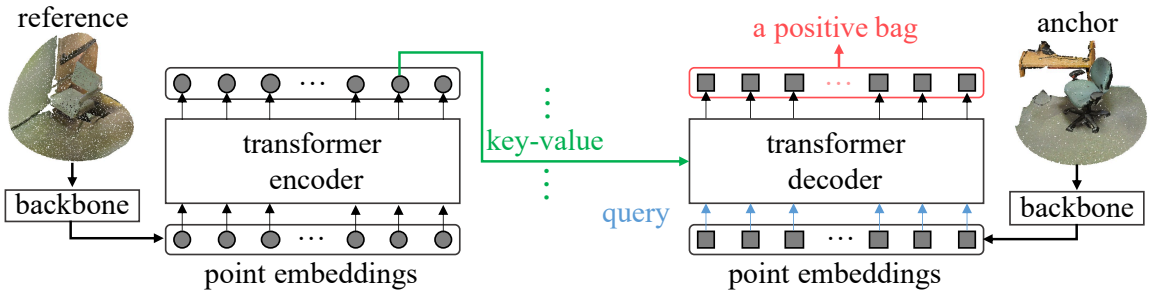


Figure 1: Given two point clouds (anchor and reference) of the same category (chair), a backbone network is applied to compute point embeddings. The transformer encoder and decoder are applied to the two clouds respectively. Self-attention captures long-range dependency. In the cross-attention module of the decoder, the points (tokens) from the anchor serve as the queries, while those from the reference act as key-value pairs. The transformer maps each query to a weighted sum of values. The outputs of the queries produce a positive bag for multiple instance learning. Once the reference is changed to another cloud without covering any chairs, the outputs of the queries then yield a negative bag for the chair category.

formation in weakly supervised segmentation, and hence is crucial to the performance. Max pooling considers only peak points, typically leading to incomplete object segments. In addition, it is sensitive to noise. Average pooling for weakly supervised segmentation often suffers from performance degradation caused by irrelevant points such as those belonging to other classes or background. Moreover, stuff categories, *e.g.* floor or wall, in point cloud segmentation bring the class imbalance problem, which makes the aforementioned issues worse. We address these issues by proposing *adaptive global weighted pooling* (AdaGWP), which introduces learnable weights, one for each class. These class-specific weights are derived so that the model can attend to points of relevant classes. It turns out that AdaGWP suppresses irrelevant points while recovering object points more completely.

We also consider cross-scale consistency of point clouds to regularize weakly supervised feature extraction. Random point sampling is applied to subsample a point cloud. Subsampling does not change point labels even if the labels are unknown in weakly supervised learning. Thus, a consistency loss is imposed to enforce the similarity between the features of the original and subsampled point clouds, acting as extra supervision signals for network training.

The main contribution of this work is the MIL-derived transformer, which explores additional inter-cloud semantics for weakly supervised segmentation. In addition, a class-specific, learnable pooling technique AdaGWP and multi-scale feature equivariance are utilized to enhance model training. Our method is flexible to work with different point cloud networks, and with diverse types of weak supervisory signals, including sparsely annotated points [42], subcloud-level [38] and scene-level [31] annotations. It performs favorably against existing methods on the S3DIS [1] and ScanNet [7] benchmarks.

2. Related work

Weakly supervised semantic segmentation on images. This task aims at reducing the expensive annotation cost of pixel-level labels for learning an image segmentation model. It works on training data with weak annotations, such as bounding boxes [18], scribbles [25], points [2], and image-level labels [20, 21, 32, 37, 39, 40, 44, 49]. Labels in the form of scribbles [25] and points [2] are referred to as *incomplete supervision*, which corresponds to partially labeled points on point clouds. These methods under incomplete supervision usually explore image-specific properties, such as spatial and color continuity [2, 25]. Image-level annotations [17, 20, 21, 23, 32, 37, 39, 40, 44, 49] are referred to as *inexact supervision*, which corresponds to subcloud-level or scene-level labels on point clouds. Many methods [20, 32, 37, 39, 40, 44] use class activation maps (CAM) [48] with classification-oriented models for object localization. Compared with methods of this category, our method is developed for point cloud segmentation. It is flexible to work under incomplete and inexact supervision.

Weakly supervised point cloud segmentation. This task learns a point cloud segmentation model under weak supervision, such as sparsely labeled points [12, 26, 42, 46, 47], subcloud-level labels [38], and scene-level labels [31]. Given sparsely labeled points *e.g.* one labeled point for each category in a scene, existing approaches use spatial constraints and different techniques, such as graph propagation [26, 42, 47], self-training [26, 47], and pre-training [12, 46], to derive segmentation models. Learning with scene-level or subcloud-level annotations is even more challenging since only class tags of clouds or subclouds are available. With scene-level annotations, Ren *et al.* [31] jointly address segmentation, proposal generation, and object detection through a cross-task consistency loss. With subcloud-level labels, Wei *et al.* [38] subsample the whole scene into subclouds and use multi-path attentions for self-training.

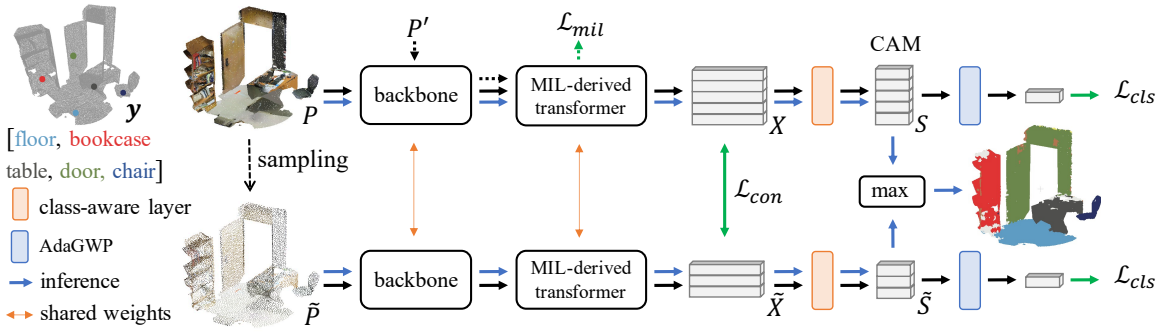


Figure 2: Overview of our method for weakly supervised point cloud segmentation. Our method integrates the three proposed components: the MIL-derived transformer, the adaptive global weighted pooling (AdaGWP), and the cross-scale feature consistency. The whole network is optimized by three loss functions, *i.e.*, $\mathcal{L}_{cls}$, $\mathcal{L}_{mil}$, and $\mathcal{L}_{con}$. The black arrows indicate the path for training, while the blue ones form the path of inference. See text for details.

Different from and complementary to the aforementioned methods, our method extends transformers to explore inter-cloud semantics for weakly supervised learning. We also present a learnable pooling technique for class-specific information aggregation and implement inter-scale feature equivariance to reach the state-of-the-art performance.

Co-segmentation and cross-image pattern mining. The co-attention module [13, 32, 43] aims at discovering the co-occurrence areas among multiple images. It has been used for object co-segmentation [10, 13, 14, 24, 45]. For instance, Hsu *et al*. [13] design a co-attention generator to consider feature discrepancy across images and produce co-segmentation maps by using contrastive learning. Sun *et al*. [32] utilize contrastive co-attention to capture cross-image semantics via computing an affinity matrix for a pair of images. Methods for object proposal or saliency map generation are usually required for common area mining among multiple images, but they are not applicable to point clouds. Our method addresses the unavailability of object proposals and saliency maps by exploring the cross-attention mechanism in transformers. We generalize the transformer [3, 34, 35] with its encoder-decoder architecture to identify inter-cloud co-occurrence points, and derive it under weak supervision by multiple instance learning.

Global and weighted pooling. Pooling is widely used to aggregate global information and handle uncertainty in weakly supervised learning. Some advanced pooling methods integrate channel-wise and spatial information [8] or include spatial attention [16]. Kolesnikov *et al*. [20] predefine the decay weight for each class before pooling. Compared with these pooling methods, the proposed AdaGWP learns a weight for each class, which is associated with a channel in CAM to suppress less relevant points. Combining the proposed MIL-derived transformer and AdaGWP, our method carries out point-specific identification and class-specific suppression simultaneously, which is essential to weakly supervised point segmentation.

3. Proposed method

This section presents the proposed method. We first give an overview of the method and elaborate on the proposed MIL-derived transformer. Then, we describe the adaptive global weighted pooling and cross-scale feature equivariance. Finally, the implementation details are provided.

3.1. Overview

We are given a weakly annotated set of N point clouds with either cloud-level labels or sparsely labeled points, *i.e.*, $D = \{P_n, \mathbf{y}_n\}_{n=1}^{N}$, where P_n denotes the nth point cloud and $\mathbf{y}_n$ is its label. Without loss of generality, we assume that each cloud has M points, *i.e.*, $P_n = \{\mathbf{p}_{nm}\}_{m=1}^{M}$, where each point $\mathbf{p}_{nm} \in \mathbb{R}^3$ is represented by its 3D coordinate. If cloud-level labels are given, $\mathbf{y}_n \in \{0,1\}^C$ is a C-dimensional binary vector indicating which categories are present in cloud P_n, where C is the number of object categories. If sparsely labeled points are provided, $\mathbf{y}_n$ records the categories of the labeled points of cloud P_n. With the weakly labeled dataset D, we aim to derive a segmentation model, which classifies each point of a testing cloud into one of the C categories or the background.

Figure 2 illustrates our method. In training, we consider a point cloud P and its label $\mathbf{y}$. A backbone network, such as 3D U-Net [6], is employed to extract the per-point embeddings. The embeddings are then fed into the MIL-derived transformer's encoder to produce the self-attention features $X = \{\mathbf{x}_m\}_{m=1}^{M}$ of P, where M is the number of points. The MIL-derived transformer's decoder is applied to another cloud P'. As shown in Figure 1, point clouds P and P' serve as the reference and the anchor, respectively. For each category c present in anchor P', the transformer outputs a positive bag if category c is also present in P, or a negative bag otherwise. An MIL (multiple instance learning) loss $\mathcal{L}_{mil}$ works on the produced positive and negative bags, and is used to train the transformer and the preceding backbone network.

The features X of the point cloud P, produced by the transformer's encoder, serve as the input to the class-aware layer [32], *i.e.* a 1×1 convolution layer with C filters, getting the class activation maps (CAM) $S \in \mathbb{R}^{M \times C}$. The proposed adaptive global weighted pooling (AdaGWP) makes the classification predictions. A classification loss $\mathcal{L}_{cls}$ is computed based on the predictions and the label $\mathbf{y}$.

As shown in Figure 2, we apply random point sampling to the point cloud P and get its subsampled point cloud $\tilde{P}$. The same process of feature extraction is used for $\tilde{P}$. The consistency loss $\mathcal{L}_{con}$ enforces feature equivariance between the common points of P and $\tilde{P}$. In sum, the whole network is optimized in a weakly-supervised manner by the following loss function

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{mil} + \mathcal{L}_{con}, \tag{1}$$

where loss $\mathcal{L}_{cls}$ is the multi-label classification loss [38, 42] under inexact supervision or the per-point classification loss [42] under incomplete supervision. The MIL loss $\mathcal{L}_{mil}$ and the consistency loss $\mathcal{L}_{con}$ will be elaborated later.

Inference. Given a testing cloud P for segmentation, we get its sub-sampled version $\tilde{P}$ by using random sampling. Both P and $\tilde{P}$ are fed into the backbone network followed by the transformer encoder to extract their features X and $\tilde{X}$. The class-aware layer maps these features X and $\tilde{X}$ to the class activation maps, S and $\tilde{S}$, respectively. To fuse the multi-scale information, we apply the nearest upsampling method [30, 38] to $\tilde{S}$. The final segmentation results are obtained by applying the element-wise max operation to S and the upsampled counterpart of $\tilde{S}$.

3.2. MIL-derived transformer

We describe how to construct the MIL-derived transformer, which generates positive and negative bags under weak supervision. As shown in Figure 2, we apply the backbone network to a point cloud of M points $P = \{\mathbf{p}_m\}_{m=1}^{M}$. Suppose that class c is present simultaneously in P and another point cloud $P' = \{\mathbf{p'}_m\}_{m=1}^{M}$. The transformer treats P and P' as reference and anchor respectively, and takes them as the input. As illustrated in Figure 1, the transformer encoder is applied to reference P. Each encoder layer comprises a self-attention module and a feed forward network (FFN). Through the encoder, the output embeddings $\{\mathbf{x}_m\}$ for points of the reference P are obtained.

The transformer decoder is composed of layers, each of which has a self-attention module, a cross-attention module, and an FFN. The decoder applies the self-attention module to anchor P' by treating the points of P' as tokens. The cross-attention module considers both the reference P and anchor P', where each point (token) of the anchor P' serves as a *query* while each point of the reference P forms a *key-value* pair. The output embeddings of anchor P' compose *a positive bag* with M *instances*, *i.e.*, $b^+ = \{\mathbf{z'}_m\}_{m=1}^{M}$, of class c, which is then used for multiple instance learning.

The transformer maps each *query* (corresponding to a point in P') to a weighted combination of the *values* (corresponding to all points of P). This property is realized by turning off residual learning in the transformer in our implementation. Since both reference P and anchor P' contain at least one point of class c, treating $b^+ = \{\mathbf{z'}_m\}_{m=1}^{M}$ as a positive bag in MIL enforces the transformer and the preceding backbone network attend to similar or matched points in P and P'. In contrast, if the reference P is changed to a point cloud without covering any points of class c, the output embeddings of the anchor P' yield a negative bag, $b^- = \{\mathbf{z'}_m\}_{m=1}^{M}$. The reason is clear: Each instance $\mathbf{z'}_m$ is a weighted sum of the values, which are derived from points not belonging to class c. Thus, instance $\mathbf{z'}_m$ must be irrelevant to class c. Treating $b^- = \{\mathbf{z'}_m\}_{m=1}^{M}$ as a negative bag helps discard the points in anchor P' that are similar to any points of reference P, namely those irrelevant to class c.

We adopt mini-batch optimization in the implementation. For every pair of point clouds in a batch, if class c is present in at least one of them, a positive or negative bag can be created for class c depending on whether c is present in the other cloud. It follows that a set of positive bags $B^+ = \{b^+\}$ and a set of negative bags $B^- = \{b^-\}$ are collected for this batch. The developed MIL loss $\mathcal{L}_{mil}$ for class c is defined by

$$\begin{aligned}\mathcal{L}_{mil}(B^+, B^-, c) = \\ \alpha \sum_{b \in B^+} -\log f_c(b) + \beta \sum_{b \in B^-} -\log\left(1 - f_c(b)\right),\end{aligned} \tag{2}$$

where $f_c(b) = \max_{\mathbf{z} \in b} f_c(\mathbf{z})$ is the probability of bag b being positive for class c, and f_c is an MLP followed by softmax which predicts whether the input embedding $\mathbf{z}$ belongs to class c. α and β are positive constants controlling the importance of positive and negative bags, respectively. Via the MIL loss $\mathcal{L}_{mil}$, each pair of training point clouds produces extra supervisory signals for optimizing the whole network.

3.3. Adaptive global weighted pooling

Most point cloud networks [29, 36] are developed to extract point-specific features from the orderless data structure. For weakly-supervised point cloud segmentation, pooling, such as global average pooling (GAP) or global max pooling (GMP), is widely adopted to aggregate point-level features to make cloud-level predictions, with which the classification loss $\mathcal{L}_{cls}$ in Eq. 1 is enabled to supervise network training. However, GAP often suffers from performance degradation caused by dominant irrelevant points, such as those belonging to categories floor or wall. GMP emphasizes just few points with peak responses, and hence is less effective to discover more the whole segments. It is

also sensitive to noise or outliers. To reduce these unfavorable issues, we propose adaptive global weighted pooling (AdaGWP), which introduces extra learnable parameters, one for each class, and can suppress irrelevant points while discovering more complete segments.

In Figure 2, the class activation maps (CAMs) of point cloud P with M points are computed by passing through the class-aware layer. Each point of P at this stage is represented as a C-dimensional vector, *i.e.*, $P = \{\mathbf{s}_m \in \mathbb{R}^C\}_{m=1}^M$, where C is the number of classes. CAMs encode the point-class relationships: Positive $\mathbf{s}_m(c)$ implies that point m probably belongs to class c, while negative $\mathbf{s}_m(c)$ means that point m is likely irrelevant to class c. Based on this property, the proposed AdaGWP introduces one learnable parameter w_c for each class c, which is derived to determine the weights of irrelevant points. Specifically, AdaGWP applied to P is a re-weighted average pooling with its output for class c computed as follows:

$$r_c = \text{AdaGWP}(\{\mathbf{s}_m(c)\}_{m=1}^M) = \frac{\sum_{m=1}^M v_m \mathbf{s}_m(c)}{\sum_{m=1}^M v_m}, \quad \text{where } v_m = \begin{cases} 1, & \text{if } \mathbf{s}_m(c) > 0, \\ \sigma(w_c), & \text{otherwise,} \end{cases} \tag{3}$$

and $\sigma(\cdot)$ is the sigmoid function. In Eq. 3, w_c is the introduced learnable parameter and its value is optimized during training. Each point m is associated with a weight v_m. For points with positive responses for class c in CAM, their weights are set to 1, meaning that all these points will be taken into consideration during pooling. For points with negative responses, they are probably irrelevant to class c, and hence suppressed by shrinking their weights from 1 to $\sigma(w_c)$. Via AdaGWP for re-weighting CAMs, the output of point cloud P for class c, namely r_c in Eq. 3, is obtained by average pooling. AdaGWP alleviates all the issues mentioned above regarding GAP and GMP by introducing few learnable parameters $\{w_c\}_{c=1}^C$, and can substantially improve the performance of weakly-supervised point cloud segmentation in the experiments.

3.4. Cross-scale feature equivariance

Multi-scale feature equivariance constraints [15, 37] offer extra supervisory signals and can enhance weakly supervised segmentation. In this work, we extend the image scaling method to the 3D point cloud domain by random point sampling. We enforce feature equivariance among cross-scale point features, where the scale of a cloud means the number of its points. For each point cloud $P = \{\mathbf{x}_m\}_{m=1}^M$ in the training set, random point sampling is applied to P for obtaining a subset of P, *i.e.*, $\tilde{P} \subset P$. The size of $\tilde{P}$ is denoted by $\tilde{M}$. We set $\tilde{M} = \gamma M$, and the sampling ratio $0 < \gamma < 1$ is given in the implementation details. As shown in Figure 2, the point-wise features $X = \{\mathbf{x}_m\}_{m=1}^M$ of P and $\tilde{X} = \{\tilde{\mathbf{x}}_m\}_{m=1}^{\tilde{M}}$ of $\tilde{P}$ are obtained via the backbone network and the transformer's encoder. The cross-scale consistency loss on the cloud P is defined by

$$\mathcal{L}_{con}(P) = \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \|\mathbf{x}_{\pi(m)} - \tilde{\mathbf{x}}_m\|^2, \tag{4}$$

where the m-th point in $\tilde{P}$ is sampled from the $\pi(m)$-th point in P.

The consistency loss $\mathcal{L}_{con}$ enforces the feature equivariance between a point cloud at two different scales. This loss offers an additional supervisory signal to regularize the weakly-supervised training process of the segmentation model. The consistency loss $\mathcal{L}_{con}$ in Eq. 4 can be generalized directly to enforce multi-scale consistency.

3.5. Implementation details

The proposed method is implemented in PyTorch. We have used DGCNN [36], KPConv [33] and 3D U-Net [6] as the feature extractor in different experimental settings. The numbers of heads, encoder layers, decoder layers, and the width of FFN in the transformer are set to 2, 2, 2, and 256, respectively. The network is optimized on a machine with eight V100 GPUs with 512 epochs. The batch size, learning rate, and weight decay are set to 32, 10^{-3}, and 10^{-4} respectively. We use AdamW [19] as the optimizer, like the previous work [3]. The parameters α and β for $\mathcal{L}_{mil}$ are set to 0.7 and 0.3 respectively. The sampling ratio γ for $\mathcal{L}_{con}$ is set to 0.8.

4. Experimental results

This section evaluates the proposed method. First, we present the datasets and evaluation metrics. We then introduce the competing methods and provide comparisons with them. Finally, we show the analysis and ablation studies for the individual components of our method.

4.1. Datasets and evaluation metrics

We conduct the experiments on two benchmark point cloud datasets, S3DIS and ScanNet. S3DIS [1] consists of six indoor areas including 272 rooms in total. Each room is scanned with RGBD sensors and represented by point clouds with XYZ coordinates and RGB values. Following the previous practice [29, 30, 36, 42], area 5 is used as the test scene. ScanNet [7] has 1,513 training scenes and 100 test scenes with 20 classes. Following the setting adopted in [33], there are 1,201 training scenes and 312 validation scenes. For both datasets, we use mean Intersect over Union (mIoU) as the evaluation metric.

4.2. Competing methods and comparisons

We compare our method with the state-of-the-art segmentation methods with different supervision settings.

Method	Pub.	Sup.	Val.	Test
PointNet++ [30]	NIPS'17	Full	-	33.9
KPConv [33]	CVPR'19	Full	-	68.4
MinkNet [6]	CVPR'19	Full	-	73.6
MPRM [38]	CVPR'20	Scene	21.9	-
WYPR [31]	CVPR'21	Scene	**29.6**	**24.0**
Ours	-	Scene	26.2	-
MPRM [38]	CVPR'20	Subcloud	43.2	41.1
Ours	-	Subcloud	**47.4**	**45.8**
SPT [46]	AAAI'21	1%	-	51.1
PSD [47]	ICCV'21	1%	-	54.7
WYPR [31]	CVPR'21	20pts	51.5	-
CSC [12]	CVPR'21	20pts	53.8	53.1
OTOC [26]	CVPR'21	20pts	52.5	-
OTOC [26]†	CVPR'21	20pts	55.1	-
Ours	-	20pts	**57.8**	**54.4**

Table 1: Quantitative results (mIoU) of several point-cloud segmentation methods with diverse supervision settings on the ScanNet dataset. "Sup." denotes the type of supervision. "Pub." gives the publication venue. † indicates iterative self-training strategy without any post-processing.

Method	Pub.	Sup.	Test
PointNet++ [30]	NIPS'17	Full	53.5
KPConv [33]	CVPR'19	Full	70.6
MinkNet [6]	CVPR'19	Full	65.4
MPRM [38]	CVPR'21	Scene	10.3
WYPR [31]	CVPR'21	Scene	**22.3**
Ours	-	Scene	12.9
Xu *et al*. [42]	CVPR'20	0.2%	44.5
Xu *et al*. [42]	CVPR'20	10%	48.0
SPT [46]	AAAI'21	0.02%	45.8
PSD [47]	ICCV'21	0.02%	48.2
OTOC [26]	CVPR'21	0.02%	42.9
OTOC [26]†	CVPR'21	0.02%	43.7
Ours	-	0.02%	**51.4**

Table 2: Quantitative results (mIoU) of several point-cloud segmentation methods with diverse supervision settings on the S3DIS dataset. "Sup." denotes the type of supervision. "Pub." gives the publication venue. † indicates iterative self-training without any post-processing.

First, fully supervised methods [6, 30, 33] for point cloud segmentation are compared and they offer the potential performance upper bounds. Second, the segmentation methods [31, 38] utilizing 3D weak labels that only indicate the appeared classes in either scene or subcloud data, are compared. This type of supervision is challenging for the large-scale point cloud datasets and shows great room for performance improvement. Third, the methods using sparsely labeled points are compared [12, 26, 46, 47] by using the 20 labeled points per scene provided from official ScanNet [7, 26] benchmark. For a fair comparison, the same backbones, data pre-processing and training strategies as the state-of-the-art methods are used.

Table 1 and Table 2 report the mIoU results of the competing methods using different types of supervision. For ScanNet (Table 1), our method often considerably outperforms the existing methods, by using all different types and numbers of sparse labels. Our method without extra post-processing or iterative re-training outperforms the state-of-the-art method OTOC [26] under the same training process. It is worth mentioning that OTOC introduces several mechanisms to achieve better performance, such as pseudo labeling propagation, iterative self-training, and prediction refinement. For a fair comparison, we compare our method to OTOC with self-training only. Moreover, OTOC [26] relies on at least few annotated points for robust graph propagation, and it cannot be directly applied with subcloud-level or scene-level annotations. In addition, our method using 20 labeled points can achieve better performance than PSD [47] using 1% labeled points.

For scene-level or subcloud-level annotations, our method consistently and significantly outperforms MPRM [38] by 4.3 and 4.2 in terms of the validation mIoU, respectively. Although our method is inferior to WYPR [31] with the scene-level annotation, WYPR requires the extra 3D object proposals and shows inferior generalization results with sparsely labeled annotations. For S3DIS (Table 2), our method also often outperforms other methods. For example, by using 0.02% labeled points, the proposed method already performs favorably against Xu *et al*.'s method [42] with 10% annotations.

Figure 3 and Figure 4 show examples of the qualitative results using different types of supervisions and the comparisons with the competing methods. For subcloud-level supervision, most objects are accurately classified by our method, *e.g*., cabinets (in blue) and chairs (in gold) are classified more correctly by our method but sometimes misclassified by MPRM, showing the effectiveness of the proposed MIL transformer. We believe the formulated positive and negative bags help the network learn more discriminative features using weak supervision. Moreover, the proposed sampling consistency loss and adaptive pooling benefit the precise segmentation of the contour of objects. Our method can distinguish the objects even if they are closed, while MPRM often fails to separate the closed objects. For sparsely labeled supervision, we have similar observations as found in subcloud-level supervision. Our method with MIL-derived transformer and AdaGWP generally classifies the objects more accurately and generates much smoother segmentation results compared with the OTOC method [26].

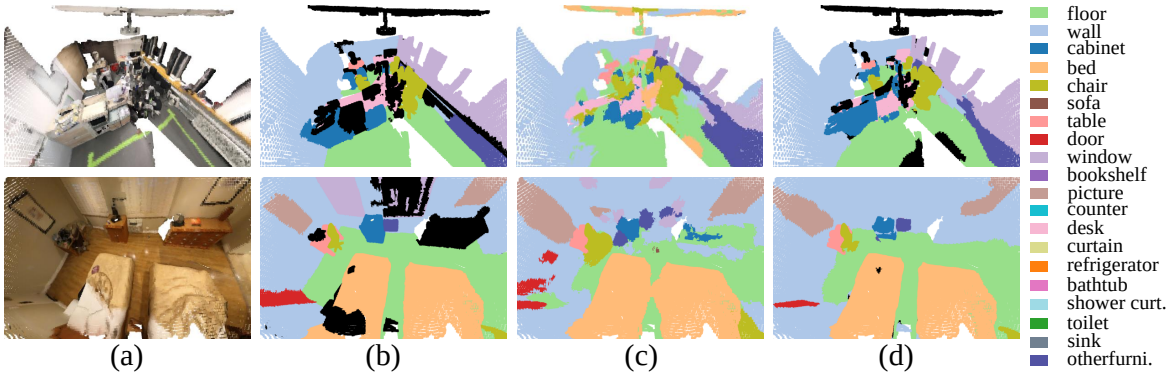


Figure 3: Examples of segmentation results on the ScanNet dataset under subcloud-level supervision. (a) Input point cloud, (b) Ground truth, (c) MPRM [38], (d) Ours. Our method provides more accurate segmentation than MPRM.

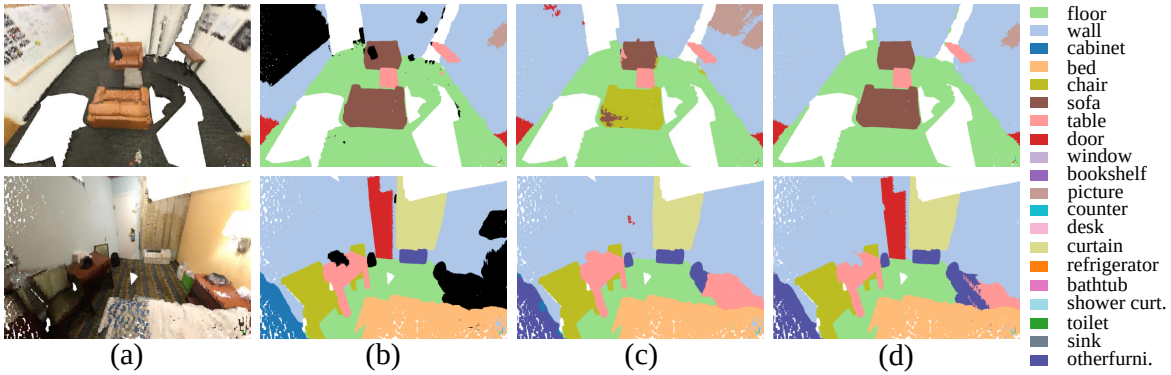


Figure 4: Examples of segmentation results on the ScanNet dataset under sparsely labeled point supervision. (a) Input point cloud, (b) Ground truth, (c) OTOC method [42], (d) Ours.

4.3. Ablation study and analysis

We report ablation studies to evaluate the effects of our proposed components and present performance analysis.

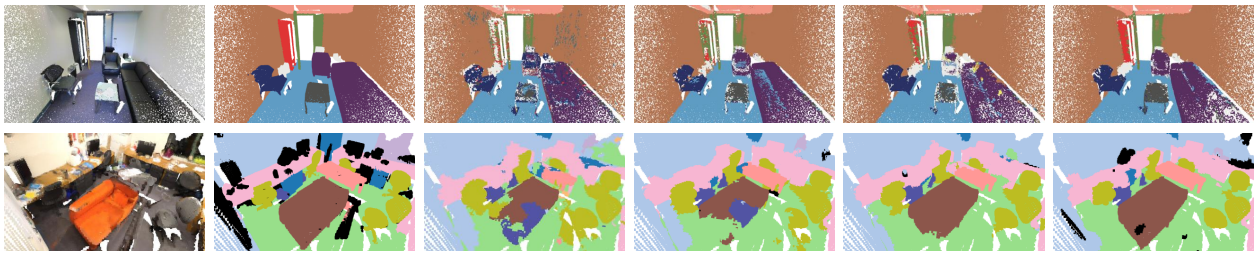
4.3.1 Contributions of components

To evaluate the effectiveness of each proposed component, we first build the baseline by considering only the class activation maps derived from standard classification loss [38, 42]. Then, we assess the contributions of the three proposed components, including the MIL-derived transformer ($\mathcal{L}_{mil}$), cross-scale consistency ($\mathcal{L}_{con}$), and AdaGWP, by adding them to the baseline one after the other. Table 3 shows the performance of different combinations with these components. The results validate that each component has its contribution. In addition, they show the generalization and effectiveness on different datasets. Finally, to show that the performance gain does not come from the transformer itself, we enhance the baseline by augmenting it with the transformer's encoder layer. The last row of Table 3 (base.+transformer) reports the performance of the enhanced baseline. It shows that the proposed MIL formulation also contributes to the performance.

Component			ScanNet	S3DIS
$\mathcal{L}_{mil}$	$\mathcal{L}_{con}$	Ada.		
base.			52.3	46.3
✓			55.4	49.1
✓	✓		55.9	49.6
✓	✓	✓	57.8	51.4
base.+transformer			55.0	47.9

Table 3: Performance in mIoU of different combinations of proposed components, including $\mathcal{L}_{mil}$, $\mathcal{L}_{con}$, and AdaGWP (Ada.), under the sparsely labeled point supervision.

Figure 5 gives the segmentation examples using our method with different combinations of the proposed components. When the MIL-derived transformer ($\mathcal{L}_{mil}$) is included, our method successfully identifies the sofa in the middle, which is misclassified as a chair by the baseline method. It is because the MIL-derived transformer helps learn better features by exploring extra intra-class and inter-class information. Both the cross-scale consistency loss ($\mathcal{L}_{con}$) and AdaGWP helps on completing objects and delineating finer object boundaries.



(a) (b) (c) (d) (e) (f)

Figure 5: Ablation study of each components on the S3DIS and ScanNet datasets. (a) Input point clouds, (b) Ground truth, (c) baseline, (d) $\mathcal{L}_{mil}$, (e) $\mathcal{L}_{mil}$ + $\mathcal{L}_{con}$, (f) $\mathcal{L}_{mil}$ + $\mathcal{L}_{con}$ + AdaGWP (our full model).

	Scene	Subcloud	20pts	Full
mIoU	26.2	47.4	57.8	73.3
label cost	$<$ 1 min	3 min	2 min	22.3 min

Table 4: Average annotation time per scene on ScanNet.

4.3.2 Performance analysis

Performance with different types of annotations. Annotating 3D point clouds is time-consuming and labor-intensive. According to previous work [26, 31, 41], the annotation cost for point cloud segmentation is 22.3 minutes per scene on average in ScanNet. To save the cost, several types of weak supervision have been introduced. As shown in Table 4, the annotation time is reduced from 22.3 minutes to 2 $\sim$ 3 minutes using sparsely labeled points [26] and subcloud-level labeling [38], and to less than a minute using scene-level labeling. Our method can work with diverse weak supervision, with the results reported in Table 4. Our method with sparsely annotated points performs significantly better than with scene-level or subcloud-level annotations. Considering the performance by using fully annotated training data, our weakly supervised method can greatly save the annotation cost.

Performance with the different parameters. The proposed MIL-derived transformer in Section 3.2 generates positive and negative bags. The hyperparameters α and β are introduced to control the relative importance between the two types of bags. Table 5 shows the performance of our method with different values of α and β, showing that the positive bags and negative bags are complementary, but the former contributes more than the latter. In Section 3.4, the cross-scale consistency loss $\mathcal{L}_{con}$ works with hyperparameter γ, the sampling ratio. We evaluate our method by setting γ to 0.25, 0.5, and 0.75, and get the performance 55.5, 56.4 and 57.8, respectively.

Performance with different pooling strategies. In Section 3.3, AdaGWP is developed to aggregate information from relevant points. We compare it with existing pooling strategies, including GMP, GAP, the one proposed by Ilse *et al*. [16], and parametric ReLU [11] followed by GAP.

MIL α	MIL β	ScanNet	S3DIS
1	0	57.4	49.1
0.7	0.3	**57.8**	**51.4**
0.3	0.7	57.1	51.2
0	1	56.9	48.9

Table 5: Performance with different values of α and β under sparsely labeled point supervision.

Pooling	ScanNet	S3DIS
GMP	52.4	46.2
GAP	55.9	49.6
Ilse *et al*. [16]	56.5	49.9
PReLU + GAP	57.1	49.8
AdaGWP	**57.8**	**51.4**

Table 6: Performance with different pooling strategies under sparsely labeled point supervision.

For comparison, our method works with each of the pooling strategies. In Table 6, AdaGWP performs favorably against all competing pooling strategies on both datasets.

5. Conclusion

This paper presents a novel method for weakly supervised point cloud segmentation. As the key component, the proposed MIL-derived transformer explores additional cross-cloud supervisory signals to facilitate weakly supervised learning, and is learned via multiple instance learning. Also, we develop cross-scale consistency and adaptive weighted pooling to improve the performance further. All proposed components are integrated into an end-to-end trainable network. Experiments show that our method outperforms existing weakly supervised methods, and even defeats some fully supervised methods. One limitation of our method is the performance bounded by the backbone network. Another limitation is that we have not fully utilized the information conveyed in the annotated points. Nevertheless, in addition to boosting the performance of weakly supervised semantic segmentation for point clouds, we believe that the proposed techniques could benefit other recognition tasks for point clouds and images with weak supervision.

Acknowledgments. This work was supported in part by the Ministry of Science and Technology (MOST) under grants 109-2221-E-009-113-MY3, 110-2628-E-A49-008, 111-2634-F-007-002, 110-2634-F-002-051, and 110-2634-F-006-022. It was also funded in part by Qualcomm and MediaTek.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *CVPR*, 2016. 2, 5

[2] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3, 5

[4] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl Wellington. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine*, 2020. 1

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *CVPR*, 2017. 1

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 1, 3, 5, 6

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *ICCV*, 2017. 2, 5, 6

[8] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *CVPR*, 2017. 3

[9] Nikolas Engelhard, Felix Endres, Jürgen Hess, Jürgen Sturm, and Wolfram Burgard. Real-time 3D visual SLAM with a hand-held RGB-D camera. In *Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, 2011. 1

[10] Junwei Han, Rong Quan, Dingwen Zhang, and Feiping Nie. Robust object co-segmentation using background prior. *TIP*, 2018. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 8

[12] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, 2021. 2, 6

[13] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Co-attention CNNs for unsupervised object co-segmentation. In *IJCAI*, 2018. 1, 3

[14] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. DeepCO3: Deep instance co-segmentation by co-peak search and co-saliency detection. In *CVPR*, 2019. 3

[15] Zeyi Huang, Yang Zou, Vijayakumar Bhagavatula, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. In *NIPS*, 2020. 5

[16] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *ICML*, 2018. 3, 8

[17] Hoel Kervadec, Jose Dolz, Meng Tang, Eric Granger, Yuri Boykov, and Ismail Ben Ayed. Constrained-cnn losses for weakly supervised segmentation. *Medical image analysis*, 2019. 2

[18] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 5

[20] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 2, 3

[21] Suha Kwak, Seunghoon Hong, and Bohyung Han. Weakly supervised semantic segmentation using superpixel pooling network. In *AAAI*, 2017. 2

[22] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018. 1

[23] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *CVPR*, 2021. 2

[24] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018. 3

[25] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 2

[26] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In *CVPR*, 2021. 1, 2, 6, 8

[27] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *NIPS*, 1997. 1

[28] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. ImVoteNet: Boosting 3D object detection in point clouds with image votes. In *CVPR*, 2020. 1

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 4, 5

[30] 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. 4, 5, 6

[31] Zhongzheng Ren, Ishan Misra, Alexander G Schwing, and Rohit Girdhar. 3d spatial recognition without spatially labeled 3d. In *CVPR*, 2021. 1, 2, 6, 8

[32] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 1, 2, 3, 4

[33] 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 *CVPR*, 2019. 5, 6

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 1, 3

[35] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 3

[36] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *TOG*, 2019. 4, 5

[37] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 2, 5

[38] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3D semantic segmentation on point clouds. In *CVPR*, 2020. 1, 2, 4, 6, 7, 8

[39] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017. 2

[40] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, 2018. 2

[41] Florian Wirth, Jannik Quehl, Jeffrey Ota, and Christoph Stiller. PointAtMe: efficient 3D point cloud labeling in virtual reality. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019. 8

[42] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, 2020. 1, 2, 4, 5, 6, 7

[43] Cheng-Kun Yang, Yung-Yu Chuang, and Yen-Yu Lin. Unsupervised point cloud object co-segmentation by co-contrastive learning and mutual attention sampling. In *ICCV*, 2021. 3

[44] Yazhou Yao, Tao Chen, Guo-Sen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhenmin Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *CVPR*, 2021. 2

[45] Ze-Huan Yuan, Tong Lu, and Yirui Wu. Deep-dense conditional random fields for object co-segmentation. In *IJCAI*, 2017. 1, 3

[46] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *AAAI*, 2021. 1, 2, 6

[47] Yachao Zhang, Yanyun Qu, Yuan Xie, Zonghao Li, Shanshan Zheng, and Cuihua Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *ICCV*, 2021. 1, 2, 6

[48] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2

[49] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *CVPR*, 2018. 2