Image Understands Point Cloud: Weakly Supervised 3D Semantic Segmentation via Association Learning

Tianfang Sun, Zhizhong Zhang, Xin Tan, Member, IEEE, Yanyun Qu, Member, IEEE, and Yuan Xie, Member, IEEE

Abstract—Weakly supervised point cloud semantic segmentation methods that require 1% or fewer labels with the aim of realizing almost the same performance as fully supervised approaches have recently attracted extensive research attention. A typical solution in this framework is to use self-training or pseudo-labeling to mine the supervision from the point cloud itself while ignoring the critical information from images. In fact, cameras widely exist in LiDAR scenarios, and this complementary information seems to be highly important for 3D applications. In this paper, we propose a novel cross-modality weakly supervised method for 3D segmentation that incorporates complementary information from unlabeled images. We design a dual-branch network equipped with an active labeling strategy to maximize the power of tiny parts of labels and to directly realize 2D-to-3D knowledge transfer. Afterward, we establish a cross-modal self-training framework, which iterates between parameter updating and pseudolabel estimation. In the training phase, we propose cross-modal association learning to mine complementary supervision from images by reinforcing the cycle consistency between 3D points and 2D superpixels. In the pseudolabel estimation phase, a pseudolabel self-rectification mechanism is derived to filter noisy labels, thus providing more accurate labels for the networks to be fully trained. The extensive experimental results demonstrate that our method even outperforms the state-of-the-art fully supervised competitors with less than 1% actively selected annotations.

Index Terms—Multimodal, weakly supervised, point cloud semantic segmentation.

I. INTRODUCTION

POINT clouds provide a powerful way to perceive, understand and reconstruct the complex 3D visual world. It reflects the fine-grained environmental information for detecting and recognizing objects [22], [40], [41] with accurate depth information. Central to the approach of point cloud vision, semantic segmentation plays a critical role in automatic driving, particularly in robotics and virtual reality [15], [30], [44].

Recently, fully supervised methods [11], [48], [60] have achieved promising performance in this community. One of the reasons for their success was the availability of well-annotated training sets. However, annotating point cloud data is exhausting, which prohibitively restricts its potential applications. For example, on SemanticKITTI [2], it takes on average 1.5-4.5 hours to label a single tile of a point cloud, while it includes approximately 23,201 scenes in total, which is unacceptable in real-world applications. Instead, there has been increasing interest in weakly supervised semantic segmentation† [21], [55], [56]. By reviewing recent weakly supervised methods, we find an important issue that has not been well addressed thus far, i.e., should we consider mining the supervision from only the point cloud data itself?

To address this issue, in a typical point cloud scenario, e.g., automatic driving, there usually involves camera sensors that assist LiDAR sensor (see the top of Fig. 1). In this case, some images are associated with the LiDAR scene, which provides rich color and texture information. Furthermore, we can explicitly establish a point-to-pixel transformation connecting 3D and 2D data by synchronizing and calibrating LiDAR and camera sensors. On the one hand, we can use image features to enhance the point cloud representations, yielding improved performance without any additional annotation costs. On the other hand, we are able to mine complementary supervision from the data connection.

Motivated by this, we propose a new and practical setting for image understanding point clouds. In this setting, as shown in Fig. 1, we assume that the point cloud is given or actively

†for labeling a tiny fraction of points.

1838 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 33, 2024
annotated with very few labels per category (1% or 0.1%), and the corresponding images are not labeled. With geometric transformations, the sparse labels from 3D points can be mapped to the corresponding 2D pixels, which enables us to explore their particular data merits and to learn additional complementary supervision.

However, there are two principal problems in this setting. 1) Large modality gaps but few labels. Few labels magnify the modality gap (see the left middle of Fig. 1, traffic lights), which makes the networks hard to optimize (see the blue line in the left bottom of Fig. 1, where networks trained with few labels exhibit the lowest convergence speed and worst performance). 2) Imbalanced modality capability. Annotations for pixels are much sparser than for the points (see red box in the middle of Fig. 1) which causes the performance of the 2D branch to be much weaker than that of the 3D branch (see the right bottom of Fig. 1, where 3D branch exhibits superior performance compared to the 2D on all of the datasets). Therefore, a naive combination of the two kinds of data can degrade the performance.

In this paper, motivated by the automated labeling strategy [6], [28], [52], we provide an efficient yet effective labeling approach that actively annotates a tiny part of labels to maximize the power of weak supervision.

To further mine the complementary supervision for the 3D network from the much weaker 2D network, we establish a self-training framework that includes a training phase and a pseudolabel estimation phase. During the training phase, we design an association learning method, taking advantage of supervision from image superpixels (pixels in visually similar regions usually contain similar semantic information). This approach helps the 3D branch to learn more discriminative features by encouraging the cycle consistency between 3D points and their corresponding superpixels. With the mined cross-modal correlation, the superpixel image prior is well learned, and complementary knowledge is hence explored, which enables us to address the “large modality gap but few labels” problem.

During the pseudolabel estimation phase, we design an adaptive confidence thresholding module and a feature similarity filtering module to improve the accuracy of the estimated pseudolabels. As a result, the 2D weak branch is boosted by learning the shared knowledge from the 3D branch, which helps the network to handle the “imbalanced modality capability”. The experimental results demonstrate that our approach outperforms state-of-the-art networks when trained with full supervision. Moreover, our approach achieves high consistency with these methods when trained with less than 1% of the annotations.

- We propose a new and practical cross-modal weakly supervised setting for point cloud semantic segmentation in which images are leveraged without additional annotations. In our approach, a dual-branch network with active labeling is proposed, which achieves state-of-the-art performance under both supervised and weakly supervised settings.
- We propose a new association learning module to take advantage of superpixel segmentation from images. This approach enables us to mine complementary knowledge from images regardless of the large modality gap and imbalanced modality capability, thereby enhancing the modality correlation and helping the network learn more discriminative features.
- We introduce a self-training approach, which alternatively performs parameter updating and pseudolabel estimation. A pseudolabel self-rectification approach is derived to filter out noisy labels. Consequently, both branches are boosted by learning the shared knowledge embedded in the reliable pseudolabels.

II. RELATED WORK

A. Fully Supervised Unimodal Learning

This branch of work can be divided into camera-based methods and lidar-based methods. For camera-based semantic segmentation, an end-to-end fully convolutional architecture was first designed by FCN [31] from an image classification network. Recently, significant improvements have been achieved by exploring multiscale information [8], [17], dilated convolution [9], [35] and attention mechanisms [19]. However, camera-based methods are very sensitive to lighting conditions and are, therefore, not robust in outdoor scenarios. LiDAR-based methods can be divided into point-based methods [22], [40], [41], [45], projection-based methods [11], [34], [57] and voxel-based methods [48], [53], [59]. PointNet [40] is the first point-based method proposed to extract features through multilayer perception. RandLA-Net [22] directly modifies this method to process large sparse outdoor LiDAR point clouds. However, point-based methods suffer performance degradation as the point cloud becomes increasingly sparse. Projection-based methods utilize effective projection methods such as spherical projection [11], [34], polar projection [57], or both [27] to map 3D point clouds into 2D images and leverage 2D convolution to extract features. Although the...
processing speed is much faster, quantization error cannot be avoided during dimension reduction. Voxel-based methods convert the point cloud to 3D voxels and utilize 3D convolution to extract the feature. These methods are less sensitive to the point density, and they introduce less quantization error. Owing to sparse convolution, the computational and memory consumption are largely reduced. Therefore, voxel-based methods are both effective and efficient. In this paper, we choose the SPVCNN [48] as our 3D backbone.

**B. Fully Supervised Cross-modal Learning**

To combine the merits of these two modalities, many researchers have attempted to enhance point cloud features with knowledge from images by fusing the 2D image features to their corresponding 3D features [12], [33]. To find the correspondence between points and pixels, a typical solution is to directly project the points to the image with a pre-defined calibration matrix [23], [51] or through a bird-eye-view (BEV) projection [54]. The features can be fused by direct connections [51] or other well-designed methods [23], [54], [58]. Recently, PMF [60] exploits a collaborative fusion of multimodal data in camera coordinates. Although noticeable performance gains can be achieved by these methods, they always rely on the fully supervised setting, requiring densely annotated data to provide supervision for network training and even additional labels to train the 2D network. In contrast, our method can leverage knowledge from unlabeled images and can be easily applied to a variety of weakly supervised scenarios.

**C. Weakly Supervised Learning**

In previous works the weakly supervised point cloud semantic segmentation problem has been addressed either by using a small part of labels per category/instance [29], [49], [55], [56] or by mining the consistency regulation within the point cloud itself [10], [20], [47]. For example, Zhang et al. [55] designed a point cloud colorization pretext task to assist in the weakly supervised learning process. In PSD [56], a self-supervised method was proposed to establish consistency between the original point cloud and the perturbed one. In [29], the sparse labels are propagated according to the feature similarity with a conditional random field in an iterative way. These aforementioned methods make good use of the knowledge from the point cloud but have proven to be effective only in indoor scenes. However, due to the large differences in color availability, density, and quantity of points, these methods may not be suitable for outdoor LiDAR point clouds. Recently, SLiDR [43] was developed as a superpixel-based cross-modal contrastive learning method for outdoor scenarios. However, its image branch must be frozen to avoid collapse during training; thus, the information from images is not fully used. However, in our method, the two branches can be trained synchronistically. Moreover, our method does not require carefully designed perturbations or graph-based label propagation; however, it can handle challenging sparse outdoor LiDAR point cloud scenes where we mine the consistency supervision from images.

**III. METHOD**

Our goal is to develop a weakly supervised 3D segmentation method that borrows complementary information from images. To this end, we design a simple yet effective point cloud segmentation baseline (Sec. III-A) based on the self-training framework (Sec. III-B). In the training phase, we introduce a cross-modal association method to connect 2D and 3D visual data and to mine their supervision (Sec. III-C). In the pseudolabel estimation phase, we introduce a pseudolabel self-rectification method to filter pseudolabels (Sec. III-D).

**A. New Problem Setting and Baseline**

1) **Problem Setting:** We use \( x_{3D} \) to denote a 3D LiDAR point cloud, and each point in \( x_{3D} \) corresponds to a label \( y \in [1, 2, \ldots, C] \), where \( C \) denotes the number of categories. In addition, images \( x_{2D} \) are provided and associated with the LiDAR scene (see Fig. 2). For example, on the nuScenes [5] dataset, there are six images corresponding to a frame of the point cloud, so we can establish a perspective transformation between points and pixels to further explore the complementary information contained in these data.

To avoid extensive annotation, we assume that very few labels of point clouds are given (e.g., 1% or fewer) and that each category has at least one labeled point. 2D images \( x_{2D} \) are without any annotations. To reduce the degree of labeling effort, we explored two labeling strategies, random labeling [21], [56] [29], which randomly annotates a tiny part of the points, and active labeling, which we introduce later. Our experimental results highlight the importance of this labeling strategy, as we can achieve much better performance with fewer annotations.

As illustrated in Fig. 3, this active labeling process contains the following three steps:

1) **Ground Detection.** For each keyframe, given the provided ego-poses, we aggregate consecutive scans to obtain the accumulated point cloud (see Fig. 3 (a)). Then, we split the point cloud into uniform pillars in cylindrical coordinates. Finally, we apply the RANSAC algorithm [14] to detect the ground points in each local cell.

2) **Density Clustering.** We use a presegmentation approach, HDBSCAN algorithm [7], to accurately cluster points...
Two categories: cars and roads. Thus, one point label for each is needed. However, points in the brown cluster belong to one category (car). Therefore, only a single click image and LiDAR data, respectively. They are chosen due to the balance of performance and efficiency. In addition, we send $x_{3D}$ and $x_{2D}$ from the same scene into the networks to produce consistent representations.

2) Modality Fusion. In the middle of the 2D and the 3D encoders, we adopt the LIFusion module proposed in EPNet [23] to fuse two kinds of features from the corresponding layer. In our implementation, for a point $x_{3D} = (x, y, z, 1)^T$ with homogeneous coordinates, we compute the projected point $\tilde{x}_{3D} = (\tilde{x}, \tilde{y}, \tilde{z})^T$ in pixel coordinates by the transformation matrix. With this approach, we can find the corresponding image pixel and therefore obtain the 3D feature $F_{3D}$ and 2D feature $F_{2D}$. We fuse the features by

$$w = \sigma(h(tanh(f(F_{3D}) + g(F_{2D}))))$$

$$F_{fuse} = F_{3D} \odot w F_{2D}. \tag{1}$$

where $f$, $g$ and $h$ represent multilayer perceptron (MLP), $\sigma$ denotes the sigmoid activation function and $\odot$ denotes the concatenation. As a result, each 3D feature is enhanced by combining it with its aligned 2D feature.

3) Loss Function. We optimize the network by cross-entropy and Lovasz-softmax loss [4] on labeled data. When applying active labeling, the learning objective consists of two parts. One is cross-entropy loss and the Lovasz-softmax loss computed for sparse and propagated labeled data; the other is computed for negative labeled data and expressed as

$$\mathcal{L}_{neg} = -\frac{1}{N} \sum_{i=1}^{N} \log(1 - \sum_{c_{ij}=0} p_{ij}), \tag{2}$$

where $p_{ij}$ is the prediction logit of point $i$ for category $j$. $c_{ij} = 1$ denotes that the cluster including point $i$ contains category $j$. We refer to this process as supervised segmentation training, denoted as $\mathcal{L}_{seg}$.

B. Cross-Modal Self-Training Overview

To train this cross-modal baseline, there are two major challenges: 1) Large modality gap but few labels. When very few annotated data are given, both the 2D and 3D branches are not fully trained. Feature fusion fails to mine complementary supervision from two kinds of data. 2) Imbalanced modality capability. The annotations in the 2D branch are extremely sparse (the number of pixels is much greater than the number of points), which leads to an imbalance in the modality capability, i.e., the 3D network is much stronger than the 2D network.

| Statistics of Active Labeling | nuScenes | Waymo | SemanticKITTI |
|-------------------------------|----------|-------|---------------|
| Sparse label                  | 0.8%     | 0.3%  | 0.08%         |
| Propagated label              | 48.5%    | 21.2% | 22.7%         |
| Negative label                | 44.5%    | 76.7% | 70.6%         |

Fig. 3. Pipeline of active labeling. a) Ground detection, b) density clustering, and c) human annotation.
To address these issues, we propose a cross-modal weakly supervised 3D semantic segmentation framework, as shown in Fig. 4. We formulate this process on the self-training framework, which is composed of a training phase and a pseudolabel estimation phase. Broadly speaking, in the training phase, our goal is to find the network parameters $\theta$ that maximize the log-likelihood function:

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{c=1}^{C} \log p(x^{(i)}, y^{(c)} | \theta),$$

where $x^{(i)}$ denotes the $i$-th sample in total of $N$, and $y$ denotes the semantic label. We omit the subscript 2D/3D for simplicity.

In the pseudolabel estimation phase, we estimate the posterior probability $p(y^{(c)} | x^{(i)}, \theta)$ to generate pseudolabels from the predictions of the network. These pseudolabels are then filtered by our proposed adaptive confidence thresholding and feature similarity filtering.

The self-training process is performed by iteratively performing the training phase and the pseudolabel estimation phase. In our self-training framework, we design a cross-modal association learning approach to take advantage of image superpixel segmentation. This framework also boosts the weak 2D branch to substantially improve segmentation results. Next, we describe our self-training framework in detail.

### C. Cross-Modality Association

The goal of the training phase is to update the parameters with the available data; however, at the beginning of training, the amount of annotated data is limited. Thus, the generated pseudolabel is not accurate. In this case, the network tend to overfit and become misleading. Therefore, to take full advantage of the available data, we propose the (Cross-Modal Association Learning, CMAsL) module. Intuitively, we can explore complementary knowledge from images, and the results can in turn facilitate 3D segmentation.

As shown in Fig. 5, we select SEEDS [3], a training-free superpixel segmentation algorithm, to obtain a rough segmentation result (i.e., superpixel). Each superpixel contains a small region of the image, and the pixels in this superpixel are highly locally consistent. By taking advantage of this prior, CMAsL is designed to imagine a random walker traveling between 2D and 3D data. The walker starts from a labeled 3D point $\tilde{f}_{3D} \in \tilde{F}_{3D}$; visiting the 2D points $\tilde{f}_{2D} \in \tilde{F}_{2D}$ and end at another labeled 3D point.

To set up an appropriate loss on this walk, we encourage 2D and 3D features with the same semantic information to be close to each other, and hence, we promote the association between them. To this end, we first map the feature dimensions of 2D and 3D features to the same. Then, we define a transition
probability from $\tilde{f}_{3D}$ to $\tilde{f}_{2D}$:

$$a_{ij}^{(l\rightarrow c)} = \frac{\exp (\langle \tilde{f}_{3D}^{(i)}, \tilde{f}_{2D}^{(j)} \rangle)}{\sum_j \exp (\langle \tilde{f}_{3D}^{(i)}, \tilde{f}_{2D}^{(j)} \rangle)},$$

(4)

where $\langle \cdot, \cdot \rangle$ represents the scalar product. We denote the transition matrix from 3D features to 2D features as $A_{3D}^{(l\rightarrow c)} \in \mathbb{R}^{N_l \times N_c}$ and the transition matrix $A_{sim}^{(c\rightarrow l)} \in \mathbb{R}^{N_c \times N_l}$ as the opposite. $N_l$ and $N_c$ represent the number of labeled 3D points and 2D features of the 3D point within the superpixels, respectively. To enhance the semantic purity, the 2D features of the 3D points within the superpixel are not utilized for association if the superpixel contains 3D points belonging to different semantic classes (indicated by the sparse label). When the walker returns, the associative similarity matrix $A_{sim} \in N_l \times N_l$ can be calculated as follows:

$$A_{sim} = A_{sim}^{(l\rightarrow c)} \times A_{sim}^{(c\rightarrow l)}. \quad (5)$$

We force the two-step probability to be similar to the uniform distribution over the class labels via a Kullback-Leibler Divergence:

$$L_{walker} = KL(Y_{sim} || A_{sim}), \quad (6)$$

where

$$Y_{sim}^{(ij)} = \begin{cases} \frac{1}{|Y_{sim}|} & y_i = y_j \\ 0 & \text{else}. \end{cases} \quad (7)$$

In addition, to explore the hard samples, we add a visit regulation $L_{vis}$ to force all the points in $\tilde{f}_{2D}$ with equal probability of being visited, which can be described as follows:

$$L_{vis} = KL(V || A_{vis}), \quad (8)$$

where

$$A_{vis,j} = \sum_i a_{ij}^{(l\rightarrow c)}, \quad V_j = 1/N_s \quad (9)$$

The association loss is finally expressed as follows:

$$L_{asso} = L_{walker} + \beta_v L_{vis}, \quad (10)$$

where $\beta_v$ is a hyperparameter. In our implementation, only 3D points in the current minibatch with sparse labels are used as the start of association learning, as they are more representative both of spatial distribution and of categorical distribution. Although propagated labels are almost as accurate as sparse labels, performing associations with them is computationally prohibitive.

In summary, CMAsL is designed to investigate the supervision for 3D features derived from neighboring 2D features within the corresponding superpixel. Such 2D features are often highly spatially and semantically correlated with 3D features, resulting in an enhancement of the generality of the 3D feature. The mined cross-modal correlation enables the effective learning of the superpixel prior and the exploration of complementary knowledge, even in the presence of a significant modality gap. No additional supervision is required.

### D. Estimating Pseudolabels via Confidence Self-Rectification

In the pseudolabel estimation phase, we generate reliable pseudolabels via a self-rectification mechanism, thereby helping the two branches to be better optimized. Therefore, we propose our (Adaptive Confidence Threshold, ACT) module and (Feature Similarity Filter, FSF) module to improve the reliability of the pseudolabels. Let us note that as the sparse labels and the propagated labels are already accurate, we generate pseudolabels from only negative labeled points and unlabeled points.

1) **Adaptive Confidence Threshold:** The training dataset is highly class imbalanced, and the network tends to be confident in the dominant classes. Thus, a fixed confidence threshold may either filter out too many accurate pseudolabels from minor classes or may preserve many noisy pseudolabels from major classes. Therefore, we propose an adaptive confidence threshold to set different thresholds for different classes. Specifically, given the predicted class distribution $p \in \mathbb{R}^{N \times C}$ and the raw pseudo one-hot labels $y_{raw}$, the confidence threshold for each category is calculated as follows:

$$\sigma_c = \max(\max(p_c) - \delta, \alpha), \forall c \in y_{raw}. \quad (11)$$

where $p_c \in \mathbb{R}^N$ denotes a vector with elements representing the $c$-th category prediction for all $N$ samples. $\delta$ and $\alpha$ are two hyperparameters representing the confidence tolerance and confidence bottom line, respectively. With $\sigma_c$, a tighter threshold is set for the categories with which the network is confident (for example, road, vegetation that enjoys more annotation points), and a looser threshold for hard categories (e.g. pedestrian, bicycle, which rarely have annotation points). As a result, ACT maintains a good balance between the accuracy and quantity of the pseudolabels.

2) **Feature Similarity Filter:** It is natural to reduce the misleading pseudolabels by using the nearest class prototype, as points from the same category are usually located closer in the embedding space. Therefore, when the predictions from the classifier and the nearest prototype conflict, the labels of the points are filtered out. Specifically, given class prototypes $P_c$ (averaging the feature of points with the same label), the predicted prototype label $\bar{y}_i^p$ of point $x_i$ is defined as follows:

$$\bar{y}_i^p = \arg \max_c \exp(f(x_i), P_{y_i}) \sum_c \exp(f(x_i), P_c), \quad (12)$$

where $f(x_i)$ is the embedding feature. Note that $\bar{y}_i^p$ also passes through our adaptive confidence threshold module to filter the noisy label. Afterward, when $\bar{y}_i^p \neq \hat{y}_i^p$, where $\hat{y}_i^p$ is the prediction given by the classifier, we discard this label.

During the first iteration of the pseudolabel estimation phase, pseudolabels are exclusively generated from the negative labeled points. In the following iteration, we generate pseudolabels from both the negative labeled points and the unlabeled points. With more accurate labels, the weak 2D network is hence boosted and can provide more useful information. The overall procedure is summarized in Algorithm 1.

### IV. Experiments

In this section, we perform experiments to present a comprehensive evaluation of our approach.
Algorithm 1 Overall Procedure

Input: Model $\mathcal{M}$
- Sparse labeled dataset $\mathcal{D}^s$
- Propagated labeled dataset $\mathcal{D}^p$
- Negative labeled dataset $\mathcal{D}^n$
- Unlabeled dataset $\mathcal{D}^u$
- Superpixel $S$

Output: Trained model $\mathcal{M}$

1: compute $\mathcal{L}_{seg}$ on $(\mathcal{D}^s \cup \mathcal{D}^p \cup \mathcal{D}^n)$
2: compute $\mathcal{L}_{asso}$ on $\mathcal{D}^s$ and the matched superpixel $S'$
3: update $\mathcal{M}$ with $\mathcal{L}_{seg}$ and $\mathcal{L}_{asso}$
4: repeat
5: generate pseudo labels $\mathcal{D}'$ using $\mathcal{M}$
6: compute $\mathcal{L}_{seg}$ on $(\mathcal{D}^s \cup \mathcal{D}^p \cup \mathcal{D}^n \cup \mathcal{D}')$
7: compute $\mathcal{L}_{asso}$ on $\mathcal{D}^s$ and the matched superpixel $S'$
8: update $\mathcal{M}$ with $\mathcal{L}_{seg}$ and $\mathcal{L}_{asso}$
9: until The mIoU of $\mathcal{M}$ starts to degrade
10: return $\mathcal{M}$

### TABLE II

Statistics of Evaluation Datasets

| Sensor       | LiDARs | SemanticKITTI | Waymo |
|--------------|--------|---------------|-------|
| LiDARs Images| 1      | 1             | 1     |
| Avg Points/Frame | 34K    | 120K          | 177K  |
| Annot.       |        |               |       |
| Training Set | 28,130 | 19,132        | 23,691|
| Validation Set | 6,019  | 4071          | 5,976 |
| Subset       |        |               |       |
| Training Set | 76.8%  | 15.9%         | 64.4% |
| Validation Set | 76.8%  | 16.3%         | 64.3% |

A. Experimental Setting

1) Datasets: We empirically evaluate the performance of our method on benchmark datasets, including nuScenes [5], SemanticKITTI [2] and Waymo [46] datasets. The basic statistics of the datasets are summarized in Tab. II. nuScenes contains 1,000 driving scenes collected from two different cities, Boston and Singapore, with different weather and light conditions. The scenes are 20 s each and are split into 28,130 training frames and 6,019 validation frames. Each frame contains a 32-beam LiDAR point cloud provided with pointwise annotations and six RGB images captured by six cameras from different views of LiDAR. Sixteen categories are used for segmentation. SemanticKITTI is a large-scale dataset based on the KITTI Odometry Benchmark [16] captured in Germany. A total of 43,000 scans with pointwise semantic annotations are provided, where 23,201 scans (sequence 00-10) are available for training (19,130 scans) and validation (4,071 scans). Unlike nuScenes, SemanticKITTI provides only front-view images. Nineteen categories are used for segmentation. Waymo contains 2,030 segments of 20 s each captured from Phoenix, San Francisco, and Mountain View, with diverse geographies and conditions. The segments are split into 800 training segments (23,691 frames in total) and 350 validation segments (5,976 frames in total). Each frame contains a high-resolution LiDAR point cloud captured by five LiDAR sensors (a mid-range LiDAR and four short-range LiDARs; we use only the points captured by the mid-range LiDAR) with pointwise annotations and five RGB images captured by five cameras from the front and side views of the car. Twenty-two categories are used for segmentation.

2) Cross-modal Data Processing: To ensure that every 3D point has its corresponding 2D pixel, following [60], we remove all the points that are out of the view of the cameras and therefore construct the corresponding subsets for training and evaluation. In fact, these subsets retain almost the same number of points for both the training and validation sets. Therefore, they reflect the entire data distribution. For example, as shown in Tab. II, for both the training and the validation sets, approximately 80% of the data in nuScenes are included in the subset. To make a fair comparison, all of the released models of the state-of-the-art methods are trained and evaluated under their original datasets and settings, but we collect the predictions of the points only within our evaluation subset to calculate the final performance.

3) Evaluation Protocols: Following prior work [40], we use the mean intersection-over-union (mIoU) as the main evaluation metric. We report the results under fully supervised and weakly supervised settings. If not specified, the baseline is trained with only the sparse label. In addition, we compare our method with random labeling in the ablation study, where we randomly select the same number of points for annotation. Additionally, to eliminate the influence of the backbone network on the performance of each weakly supervised method, we further report their relative performance against the fully supervised method, denoted as WS/FS.

4) Iteration Protocols: For active labeling, in the first iteration, we estimate pseudolabels only at the points that are marked as negative labels. In the following iteration, we resume the pseudolabel operation in the previous iteration and generate new pseudolabels from the remaining negative labels and unlabeled points.

B. Implementation Details

We use SPVCNN and SwiftNet as the backbones for the LiDAR stream and camera stream, respectively. For nuScenes and Waymo, we initialize the parameters of SwiftNet with ResNet-18 [38] pretrained on ImageNet. For SemanticKITTI, we adapt SwiftNet to ResNet-34 and fine-tune it after pre-training on nuScenes, as SemanticKITTI includes only one camera, which is insufficient for training. We adopt SGD with Nesterov [36] as our optimizer for both 3D and 2D networks. For nuScenes, Waymo, and SemanticKITTI, the initial learning rates for the 3D branch and fusion module are 0.24, 0.24, and 0.18, respectively. The initial learning rates for the 2D branch are 0.24, 0.004, and 0.0002. All decay to 0 with the cosine rule [32]. The batch size for all of the datasets is set to 12. Given that the SPVCNN is based on 3D voxels, we set the voxel size to 0.05 m. The hyperparameters $\alpha$ and $\delta$ in ACT are set to 0.5 and 0.1, respectively. The weights for $\mathcal{L}_{seg}$, $\mathcal{L}_{asso}$ and $\mathcal{L}_{vis}$ are set to 1.0, 0.5 and 0.5, respectively. To implement the association learning, we use two projection heads to align the point cloud features and image embeddings, where the output dimension is set to 256.
To prevent overfitting, we applied random rotation along the z-axis and random scaling on 3D point clouds.

### C. Results on nuScenes

We report the quantitative results on nuScenes in Tab. III. Even with 0.8% of the points being sparsely labeled, our method outperforms almost all the recent fully supervised methods. Specifically, under the fully supervised setting, our baseline with image fusion outperforms the state-of-the-art LiDAR-only methods such as RPVNet [53] by 1.2% in terms of the mIoU. This indicates that the information contained in images can complement and assist LiDAR data well.

Our method also outperforms the prior cross-modal method PMF [60] by 1.9% in mIoU, demonstrating our baseline can well combine the information between 3D voxels and images. Under the 0.8% label setting, our baseline achieves 73.9%@mIoU. However, by leveraging our active labeling method and cross-modal self-training framework, our baseline receives a 3.6% performance gain and achieves 77.5% in mIoU, which is highly comparable with the state-of-the-art supervised methods.

To evaluate the weakly supervised performance of our method, we also compare our method to the most related work, SLiDr [43], which also utilizes the idea of the superpixel. This process is performed via label-agnostic contrastive learning. The main difference is that 1) SLiDr freezes most of the weight of the 2D branch during training, while these weights are updated in our method, which allows us to train the whole network jointly; 2) in contrast to contrastive learning, our self-training strategy consistently improves the performance. Therefore, our learned image feature is more compatible with the 3D branch. As the label setting of SLiDr is different from ours, to make a fair comparison, we adapt SLiDr by training it with the same percentage of randomly sampled labels. The class distribution of the label is guaranteed by proportionally selecting the label from each class. As shown in Tab. III, with the same amount of annotation, our method surpasses SLiDr by 7.4@mIoU and 4.4@WS/FS, demonstrating the effectiveness of our cross-modal self-training framework.

### D. Results on SemanticKITTI

As shown in Tab. IV, we report our quantitative results on SemanticKITTI. We note that this dataset provides only front-view images; thus, we have to use only the front-view point cloud for training and testing, following prior cross-modality work [60]. Even in this situation, our model still achieves a 64.9@mIoU that is 1% greater than that of PMF [60] and only 1% lower than that of the state-of-the-art lidar-only method Cylinder3D [59], which is trained on the full training set. To make a fair comparison, we pretrain PMF on the nuScenes training set with full labels and then fine-tune the network on SemanticKITTI, as in our method. However, we obtain only 61.8% in terms of the mIoU, as shown in Tab. IV PMF*. Additionally, many existing weakly supervised methods often adopt RandLANet as the baseline, and its performance is somewhat weak. Owing to our strong fusion baseline, our approach outperforms those methods by a large margin. We also compare our method with recently proposed voxel-based backbones. For example, Scribble [50] adopted scribbles for LiDAR point clouds; these scribbles are popular in 2D semantic segmentation and achieved a 61.9@mIoU with 8% annotated points. ReDAL [52] employs active labeling, reaching 59.8@mIoU with 5% annotation. Our method surpasses all of these methods by achieving 63.7@mIoU and 98.2@WS/FS with the least annotation of 0.08%. Compared with that of our baseline method trained only with sparse labels, our full method achieves a 3.5@mIoU performance improvement. These experimental results also support our claim that combining LiDAR data and images can be beneficial, especially in a weakly supervised setting.

### E. Results on Waymo

As shown in Tab. V, we further report our quantitative results on Waymo. Given that it is a recently released dataset,
the official results of most methods are unavailable. Therefore, we report the results by reimplementing the methods with officially released codes. The key observations are as follows: 1) Our baseline achieves the highest performance in a fully supervised manner. Like for nuScenes, our method outperforms SPVCNN by 1.5@mIoU by using complementary information from images. 2) Our method also outperforms the cross-modality method PMF by a large margin. We find that PMF relies on projecting sparse LiDAR point clouds to dense perspective view images, but most of the pixels are empty on Waymo, which leads to a suboptimal solution. 3) Under the weakly supervised setting, we obtain 2.2@mIoU and 3.3@WS/FS gains due to the self-training framework.

**F. Ablation Study**

In this subsection, we conduct ablation studies to further verify the effectiveness of each component of our method. For a fair comparison, we follow the same experimental settings with 0.8% active labeling on nuScenes. The official results of most methods are unavailable. Therefore, we report the results by reimplementing the methods with officially released codes. The key observations are as follows:

1. Our baseline achieves the highest performance in a fully supervised manner. Like for nuScenes, our method outperforms SPVCNN by 1.5@mIoU by using complementary information from images. 2. Our method also outperforms the cross-modality method PMF by a large margin. We find that PMF relies on projecting sparse LiDAR point clouds to dense perspective view images, but most of the pixels are empty on Waymo, which leads to a suboptimal solution. 3. Under the weakly supervised setting, we obtain 2.2@mIoU and 3.3@WS/FS gains due to the self-training framework.

### TABLE IV
**QUANTITATIVE RESULTS OF THE VARIOUS APPROACHES ON SEMANTIC KITTI VALIDATION SET**

| Methods          | Annot. | car | bicycle | motorcycle | truck | other-vehicle | person | bicyclist | skateboard | traffic-sign | vegetation | truck | rta | pej | mIoU(2D) | WtSFS(%) |
|------------------|--------|-----|---------|------------|-------|---------------|--------|-----------|------------|-------------|------------|-------|----|----|--------|---------|
| RaGNet++ [19] [34] | 100%   | 94  | 26.5    | 48.4       | 35.9  | 26.7          | 54.8   | 89.4      | 92.9       | 37.0        | 69.9       | 0     | 83.4| 51.0| 83.3| 54.0    | 68.1    |
| RaGNet++ [20] [34] |        | 95  | 8.0     | 12.8       | 78.8  | 46.7          | 52.3   | 46.0      | 93.4       | 33.7        | 73.4       | 0.1  | 84.0| 43.8| 83.7| 73.2    | 73.1    |
| SPVCNN [20] [48]  | 100%   | 96  | 6.2     | 32.0       | 52.9  | 35.4          | 67.3   | 82.0      | 91.9       | 30.1        | 76.0       | 1.1  | 87.5| 47.8| 84.8| 62.0    | 65.3    |
| SalsaNet [20] [48] |        | 90  | 4.6     | 49.6       | 66.3  | 54.6          | 74.0   | 81.4      | 93.4       | 40.6        | 69.1       | 0.0  | 84.5| 53.0| 83.6| 64.3    | 52.4    |
| Cylinder3D [21] [59] |      | 96  | 6.1     | 28.7       | 66.7  | 74.5          | 82.8   | 94.4      | 95.4       | 46.9        | 79.0       | 2.1  | 88.3| 53.1| 66.9| 73.2    | 69.0    |

*Results of these methods are reported by reimplementing the methods with officially released codes. † Results of these methods are calculated from the entire validation set as official codes are unavailable.*

### TABLE V
**QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON THE WAYMO VALIDATION SET**

| Methods          | Modality | Annot. | mIoU(%) | WS/FS(%) |
|------------------|----------|--------|---------|----------|
| SPVCNN [20] [48] | LiDAR    | 100%   | 65.5    | -        |
| Cylinder3D [21]  | LiDAR    |        | 62.6    | -        |
| PMF [21] [60]    | LiDAR    |        | 58.2    | -        |
| **Our Baseline**  | Camera   |        | 64.4%   | 67.0     |
| **Ours**         | Camera   | 0.3%   | 63.5    | 94.8     |

### TABLE VI
**ABLATION STUDY ON nuScenes Validation SET. SP DENOTES THE USE OF THE SPARSE LABEL. PP DENOTES THE USE OF THE PROPAGATED LABEL. NEG DENOTES THE USE OF THE NEGATIVE LABEL AND ST. DENOTES THE USE OF THE SELF-TRAINING FRAMEWORK. mIoU(2D) IS CALCULATED ON THE 3D-PROJECTED POINTS.**

| Components | mIoU(3D) | mIoU(2D) |
|------------|----------|----------|
| sp | pp | neg | CMAst | ST. | 73.3 | 58.5 |
| | | | | | 73.9 | 54.8 |
| | | | | | 74.9 | 56.8 |
| | | | | | 76.1 | 60.5 |
| | | | | | 77.1 | 61.1 |
| | | | | | 77.5 | 62.1 |

### TABLE VII
**COMPARISON BETWEEN THE VARIOUS CROSS-MODAL LEARNING METHODS. **+xMUDA and +SPCML STAND FOR ADDING THEIR CROSS-MODAL LEARNING LOSS BASED ON OUR BASELINE. +PMF STANDS FOR ADDING ITS PERCEPTION-AWARE LOSS BASED ON OUR BASELINE.

| mIoU(3D) | 76.1 | 76.0 | 76.5 | 76.1 | 77.1 |
|---------|------|------|------|------|------|
| mIoU(2D) | 60.5 | 60.9 | 60.6 | 60.2 | 61.1 |
method absorbs valuable human supervision based on the geometric prior of the LiDAR point cloud and enables us to significantly reduce labeling efforts.

As shown in Fig. 7, we randomly sample a point cloud from nuScenes to show how the point cloud is labeled with our method. Fig. 7. a, b, c, and d denote the whole active labeling results. For example, in c, we show the sparse labeled points in red, and we emphasize them with a red circle. The colored points are annotated with propagated labels. As we can see, instead of carefully labeling the entire car, only a single click can perform the same task, which can significantly save the time and annotation budget. For more examples, please refer to Sec. C in the supplementary material.

2) Effectiveness of CMAsL: As shown in Tab. VI Row (e), with the help of CMAsL, a 1.0@mIoU performance gain can be observed with our strong baseline. To further verify that our method can better transfer complementary knowledge from images to point clouds, we compare our method with related cross-modal learning methods, including xMUDA [24], DsCML [39], and PMF [60]; the results are reported in Tab. VII. xMUDA [24] employs mutual learning directly on the prediction from the 2D and 3D branches. SpCML was adapted from DsCML [39], where the feature is pooled from the corresponding superpixel if the corresponding points came from only one category. Then, we employ mutual learning on the predictions from 3D points and 2D superpixels. PMF [60] estimates entropy based on predictions from both branches and conducts predictions aligning from high-entropy modality to low-entropy modality. As shown in Tab. VII, xMUDA and PMF fail to improve the 3D branch; hence, the naive combination of the two branches degrades the performance because of the imbalanced modality capability. Although SpCML also improves the performance with the help of superpixel pooling, the improvement is slight. However, our CMAsL module offers significant benefits to the 3D branch by leveraging prior knowledge from images. Simultaneously, the 2D branch is enhanced by incorporating knowledge from the point cloud data.

3) Effectiveness of Self-training: We use the self-training framework to generate pseudolabels, and then, we use these labels to fine-tune the network. According to the last row in Tab. VI, iterative training with the ACT and FSF modules improved the performance of the 3D and 2D branches by 0.4@mIoU and 1.0@mIoU, respectively. Furthermore, we compare our method with other representative pseudolabel filtering methods. The results are shown in Fig. 6. The ESL [42] filters out the points whose entropy is larger than a given threshold. The threshold is set to be the median of the entropy of the points from each category. DARS [18] has been proven to be very effective in semi-supervised 2D semantic segmentation. The confidence threshold is set for each category to select a group of pseudolabels that maintain the same distribution as the labeled dataset. In our experiments, we also compare a fixed confidence threshold method with our ACT module, denoted as FIX. Along with all of the methods, our proposed method performs slightly better. Steady improvements can be found in every iteration. It appears that ESL hardly improves the results after two rounds of iteration, while DARS even performs slightly worse because the category imbalance issue on 3D datasets is even more significant than that on 2D datasets. This causes DARS to be very aggressive in minor classes, which introduce many noisy labels. The FIX confidence threshold method cannot either benefit from the noise introduced by the fixed threshold.

G. Robustness Analysis

In this section, we perform robustness analyses to evaluate the sensitivity of each proposed component to its corresponding hyperparameter. Notably, all these experiments were conducted on the nuScenes dataset, with consistent hyperparameters for a fair comparison.

1) Robustness to Superpixel Segmentation Performance: These experiments involve the use of SEEDS [3] with varying hyperparameters, SLIC [1], and a conventional grid segmentation approach in which each image is divided into fixed-size squares (20 × 20 pixels). The results of these experiments are presented in Tab. VIII. Let us note that all these experiments are conducted in a 0.8% annotation setting.

The key findings from these experiments are as follows: 1) Across all superpixel algorithms tested, including the grid-based segmentation, the model with our proposed CMAsL module consistently outperforms the model without it. This outcome underscores the robustness of our CMAsL module to different superpixel algorithms. 2) The SEEDS algorithm, specifically when configured with a maximum of 1024 superpixels, yields the best superpixel results, consequently leading to the highest mIoU.

In conclusion, our proposed CMAsL module effectively leverages prior knowledge encoded in superpixels and demonstrates robustness across various superpixel algorithms.

2) Robustness to the Clustering Algorithm and Number of Accumulated Frames: We conduct a series of experiments to
Fig. 7. Visualization of the active labeling results. Fig. (a) shows the presegment result where points in the same color belong to the same cluster. Fig. (b) shows the sparse label based on the presegment result, where points in black are unlabeled. Fig. (c) shows the propagated label. Fig. (d) shows the ground truth label. Fig. (e) shows the points covered by at least one of the three types of labels.

TABLE VIII

| Algorithm  | Superpixel Max Number | Avg Number | ASA(%) ↑ | Performance mIoU(3D) | mIoU(2D) |
|------------|-----------------------|------------|----------|-----------------------|----------|
| (a)        | 0                     | 0          | -        | 76.1                  | 60.5     |
| (b)        | SEEDS [3]             | 512        | 152      | 91.3                  | 78.5     |
| (c)        | SLIC [1]              | 152        | 324      | 94.7                  | 77.3     |
| (d)        | Grid                  | 376        | 376      | 91.9                  | 76.7     |

In conclusion, our approach demonstrates superior performance when employing HDBSCAN as the clustering algorithm during the active labeling process. Furthermore, our active labeling strategy is robust to changes in the number of accumulated frames and delivers improved results with denser point clouds as input.

H. Efficiency Analysis

In this section, we evaluate the efficiency of our proposed cross-modal 3D segmentation baseline. During inference, both modalities are used, which is a common practice in cross-modal point cloud semantic segmentation and detection [25], [60]. However, since predictions of the camera branch are fused into the LiDAR branch, we remove the decoder of the camera branch to speed up the inference. We report the inference latency and parameters in Tab. IX. Our proposed baseline outperforms PMF and is $2.5 \times$ faster with fewer parameters.

I. Label Efficiency Comparison

In this section, we estimate how much annotation effort can be saved by leveraging our method.
Specifically, we conducted experiments by using the nuScenes dataset, comparing our method with the traditional semi-supervised setting by measuring the mIoU at the same level of annotation time consumption. The results are reported in Tab. XI.

The key observations are: 1) Randomly annotating the same number of points per point cloud as our active labeling strategy results in an mIoU of 73.3, which is 4.2 lower than us. 2) When controlling for the same time consumption, only 15% of the training set can be fully annotated, which results in an mIoU of 68.3, which is 7.2 lower than that of our method. 3) To achieve a similar performance as the active labeling strategy, at least 40% of the LiDAR frames must be fully annotated, which costs 2.65 times more human labor than does our active labeling strategy.

The detailed estimation process is as follows: In the traditional semi-supervised setting, some of the point clouds in the training set are fully annotated, while others are unlabeled. The time consumption $T$ of our labeling strategy can be given by:

$$T = N \times r \times t,$$

where $N$ is the average number of points per point cloud, $r$ is the average label rate per point cloud and $t$ is the assumed time consumption of annotating a single point. Given that $N = 34000$, $r = 0.8\%$, and $t = 3(s)$, annotating one point cloud based on our method takes approximately 13.6 minutes. Unfortunately, the time consumption of labeling a point cloud in the nuScenes has not been officially released. Therefore, we have to take the time consumption in the SemanticKITTI dataset (1.5 hours to 4.5 hours per tile of point cloud) as an approximation. If we take the upper bound of 1.5 hours, to maintain the same time consumption, we have to fully annotate only $13.6(min)/90(min) \approx 0.151$ of the training set. Therefore, we uniformly sample 15.1% of the training set of the nuScenes dataset and train our network using this subset. Finally, the experimental results are presented in Tab. XI.

### J. Intra-Domain Generalization Analysis

In this section, we report the intra-domain generalization analysis results of our method, as some readers may be interested in whether our network trained only with the frontal view can work with side or real camera views. The results are presented in Tab. XII. Note that the other views were also used during retraining but were solely supervised by the generated pseudolabels.

Two important observations can be made: 1) The presence of an intra-domain gap and the reduction in training samples significantly affect the performance. Our network achieved an mIoU of 78.8 when trained on the full training set. However, when trained exclusively on the front view, the performance decreased to 58.3. 2) Despite the challenges posed by this disadvantageous scenario, our network still managed to improve the performance to 64.8% when using its generated pseudolabels. This outcome is reasonable because it leverages point clouds from the other views with pseudolabels.

In conclusion, our network is not restricted to the frontal view and has demonstrated its ability to manage different camera views, demonstrating its generalizability.

### K. Visualization

In this section, visualization results are provided to intuitively demonstrate the effectiveness of our approach.

1) **Cross-modal Association:** As shown in Fig. 8, we compute the feature similarity from a randomly selected labeled 3D point to the 2D superpixels (center point) by using the association projection heads. The results are shown in a heatmap. The red areas in the pictures represent high similarity, and the blue areas represent low similarity. The areas with high similarity in two kinds of data contain the same object, e.g., the road in Fig. 8. a) and c). This finding indicates that our association framework strengthens the connection between the two modalities.

2) **Pseudolabel Self-rectification:** Fig. 9 visualizes the effects of our pseudolabel self-rectification module. Fig. 9. a) and b) show the ground truth labels in the image view and point cloud view, respectively. Fig. 9. c) and d) show the estimated pseudolabels with our pseudolabel self-rectification module and with a fixed confidence filter. The black points represent filtered points. The difference is shown in the red box in Fig. 9. In d) many points are mislabeled and cannot be filtered out by the fixed confidence filter; moreover, our methods can maintain the correct estimation in Fig. 9. c). This result demonstrates that our pseudolabel self-rectification module is able to reduce the error rate of the pseudolabels.
Fig. 8. Visualization of the association results. We compute the feature similarity from a randomly selected 3D point to all 2D superpixels. The sequence from blue to red represents the similarity from low to high. a) shows the selected 3D points (the blue circle indicates a point on the road, the orange indicates a point on the wall, and the green indicates a point on the tree). b) to d) present the feature similarity of every superpixel to the points on the wall, road, and tree, respectively. e) to f) show additional association results. Note that even for small categories such as pedestrians (f) and traffic cones (h), our method can successfully build correlations between the two modalities.

Fig. 9. Visualization of the pseudolabel estimation results with and without self-rectification. a) and b) present the ground truth labels in the image view and point cloud view, respectively. c) presents the pseudolabels with our self-rectification module, and d) shows the pseudolabels with only a fixed confidence threshold filter.

Fig. 10. Qualitative comparison of nuScenes (top), SemanticKITTI (mid), and Waymo (bottom). The red circles indicate the difference between the results of our method and those of other competitors in weakly supervised settings (0.8% annotation for nuScenes, 0.08% annotation for SemanticKITTI, and 0.3% for Waymo).

3) Qualitative Results Comparison: Finally, we show the segmentation results of our method, SLidR, and our baseline in the validation sets of nuScenes (top) and SemanticKITTI (mid) and show the segmentation results of our method and our baseline in the validation set of Waymo (bottom). All in weakly supervised settings in Fig. 10. Both SLidR and our baseline perform much worse at distinguishing similar categories (e.g., car and truck, fence and building), while our method avoids such mistakes due to the better combination of 2D and 3D data.

V. CONCLUSION
In this paper, we investigate a new cross-modal weakly supervised setting for 3D segmentation and propose a
cross-modal baseline. With this baseline, we design a self-training solution with several critical improvements. The final framework outperforms all the other state-of-the-art methods by a large margin on three popular datasets. We believe that this new setting has great potential for further exploration in 3D segmentation. It is promising that more efficient self-training methods can be designed with better representation power and less computational cost.

REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[2] J. Behley et al., “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9296–9306.
[3] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, “S3RENS: Superpixels extracted via energy-driven sampling,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2012, pp. 13–26.
[4] M. Berman, A. R. Triki, and M. B. Blaschko, “The Lovasz–Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4413–4421.
[5] H. Caesar et al., “NuScenes: A multimodal dataset for autonomous driving,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11618–11628.
[6] L. Cai, X. Xu, J. H. Liew, and C. Sheng Foo, “Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 10983–10992.
[7] R. J. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” in Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining, Cham, Switzerland: Springer, 2013, pp. 160–172.
[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018.
[9] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, arXiv:1706.05587.
[10] Y. Chen, X. Tan, Z. Zhang, Y. Qi, and Y. Xie, “Beyond the label itself: Latent labels enhance semi-supervised point cloud panoptic segmentation,” 2023, arXiv:2312.08234.
[11] T. Cortinhal, G. Tzelpis, and E. E. Aksoy, “Salsanext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds,” in Proc. Int. Symp. Vis. Comput. Cham, Switzerland: Springer, 2020, pp. 207–222.
[12] K. El Madawi, H. Rashed, A. El Sallab, O. Nasr, H. Kamel, and Y.-G. Jiang, “MSMDFusion: Fusing LiDAR and camera at multiple scales with multi-depth seeds for 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2023, pp. 21643–21652.
[13] M. Li et al., “HybridCR: Weakly-supervised 3D point cloud semantic segmentation via hybrid contrastive regularization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 14910–14919.
[14] V. E. Lioung, T. N. Tho Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, “AMVNet: Assertion-based multi-view fusion network for LiDAR semantic segmentation,” 2020, arXiv:2012.04934.
[15] M. Liu, Y. Zhou, C. R. Qi, B. Gong, H. Su, and D. Anguelov, “Less: Label-efficient semantic segmentation for LiDAR point clouds,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 70–89.
[16] Z. Liu, X. Qi, and C.-W. Fu, “One thing one click: A self-training approach for weakly supervised 3D semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 1726–1736.
[17] Z. Liu et al., “LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 2831–2840.
[18] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” 2016, arXiv:1608.03983.
[19] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Vallespi-Gonzalez, “Sensor fusion for joint 3D object detection and semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019, pp. 1230–1237.
[20] A. Milioš, J. Ivozi, J. Behley, and C. Stachniss, “RangeNet++: Fast and accurate LiDAR semantic segmentation,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Nov. 2019, pp. 4213–4220.
[21] Y. Mohan and A. Valada, “EfficientPS: Efficient panoptic segmentation,” Int. J. Comput. Vis., vol. 129, no. 5, pp. 1551–1579, May 2021.
[22] Y. E. Nesterov, “A method for solving the convex programming problem with convergence rate O (1/k2),” Dokl. Akad. Nauk SSSR, vol. 269, pp. 543–547, Jan. 1983.
[23] M. Oršic, I. Krešo, P. Bevandic, and S. Šegvic, “In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019, pp. 1259–1260.
[24] M. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 8026–8037.
[25] D. Peng, Y. Lei, W. Li, P. Zhang, and Y. Guo, “Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3D semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 7088–7097.
[26] J. Behley et al., “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11105–11114.
[27] T. Huang, Z. Liu, X. Chen, and X. Bai, “EPNet: Enhancing point features with image semantics for 3D object detection,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 600–619.
[28] W. Li and C.-W. Fu, “OMNI: One multi-modal network for 3D point cloud segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 12602–12611.
[29] Z. Liu, X. Qi, and C.-W. Fu, “One thing one click: A self-training approach for weakly supervised 3D semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 7088–7097.
[30] Z. Liu, Y. Zhou, C. R. Qi, B. Gong, H. Su, and D. Anguelov, “Less: Label-efficient semantic segmentation for LiDAR point clouds,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 70–89.
A. Saporta, T.-H. Vu, M. Cord, and P. Pérez, “EGL: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation,” 2020, arXiv:2006.08658.

C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet, “Image-to-LiDAR self-supervised distillation for autonomous driving data,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 8981–8989.

T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4758–4765.

H. Shuai, X. Xu, and Q. Liu, “Backward attentive fusing network with local aggregation classifier for 3D point cloud semantic segmentation,” IEEE Trans. Image Process., vol. 30, pp. 4973–4984, 2021.

P. Sun et al., “Scalability in perception for autonomous driving: Waymo open dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2443–2451.

X. Tan et al., “Positive-negative receptive field reasoning for omni-supervised 3D segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 12, pp. 15328–15344, Dec. 2023, doi: 10.1109/TPAMI.2023.3319470.

H. Tang, “Searching efficient 3D architectures with sparse point-voxel convolution,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 685–702.

A. Tao, Y. Duan, Y. Wei, J. Lu, and J. Zhou, “SegGroup: Seg-level supervision for 3D instance and semantic segmentation,” IEEE Trans. Image Process., vol. 31, pp. 4952–4965, 2022.

O. Unal, D. Dai, and L. Van Gool, “Scribble-supervised LiDAR semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 2687–2697.

S. Vora, A. H. Lang, B. Helou, and O. Beijbom, “PointPainting: Sequential fusion for 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 4603–4611.

T.-H. Wu et al., “ReDAL: Region-based and diversity-aware active learning for point cloud semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15490–15499.

J. Xu, L. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, “RPVNet: A deep and efficient range-voxel fusion network for LiDAR point cloud segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 16004–16013.

J. H. Yoo, Y. Kim, J. Kim, and J. W. Choi, “3D-CVF: Generating joint camera and LiDAR features using cross-view spatial feature fusion for 3D object detection,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 35–52.

Y. Zhang, Z. Li, Y. Xie, Y. Qu, C. Li, and T. Mei, “Weakly supervised semantic segmentation for large-scale point cloud,” in Proc. AAAI Conf. Artif. Intell., vol. 35, 2021, pp. 3421–3429.

Y. Zhang, Y. Qu, Y. Xie, Z. Li, S. Zheng, and C. Li, “Perurbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15500–15508.

Y. Zhang et al., “PolarNet: An improved grid representation for online LiDAR point clouds semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9589–9607.

Z. Zhang, Z. Zhang, Q. Yu, R. Yi, Y. Xie, and L. Ma, “LiDAR-camera panoptic segmentation via geometry-consistent and semantic-aware alignment,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 3662–3671.

X. Zhu et al., “Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 9934–9943.

Z. Zhuang, R. Li, K. Jia, Q. Wang, Y. Li, and M. Tan, “Perception-aware multi-sensor fusion for 3D LiDAR semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 16260–16270.