Weakly Supervised Segmentation on Outdoor 4D Point Clouds with Progressive 4D Grouping

Hanyu Shi, Fayao Liu, Zhonghua Wu, Yi Xu, and Guosheng Lin .

***Abstract*—Recently, some weakly supervised 3D point cloud segmentation methods have been proposed to develop effective models with minimum annotation efforts. Our previous work, W4DTS, proposes a challenging task that utilizes only 0.001% points in outdoor point cloud datasets to achieve an effective segmentation model. However, under an extremely limited annotation budget, the quality of pseudo labels generated by W4DTS is unsatisfactory, which limits the segmentation performance in such scenarios. To solve this issue, we propose a progressive 4D grouping approach to group the annotated and unannotated points across space and time, which can generate high-quality pseudo labels with very sparse annotated points. Moreover, to further improve our progressive 4D grouping approach, we design a cross-frame contrastive learning and a local consistency learning to improve the quality of our 4D grouping. Experimental results reveal that with only 0.001% annotations, our solution significantly outperforms the previous best approach on SemanticKITTI. We also evaluate our framework on the SemanticPOSS dataset and ScribbleKITTI dataset, and achieve performances close to our fully supervised backbone models.**

***Index Terms*—3D Computer Vision, Semantic Segmentation, Weakly Supervised Semantic Segmentation.**

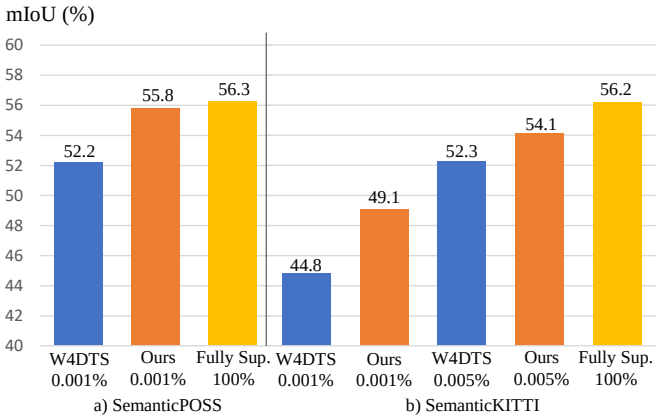


Fig. 1: **A comparison between our method and previous works on SemanticKITTI and SemanticPOSS**. We compare our method with W4DTS and the fully supervised method (Fully Sup.). The weak annotation setting of our method is the same as W4DTS.

I. Introduction

OUTDOOR 3D point cloud semantic segmentation has been attracting increasing research attention recently due to its wide applications in scene understanding, autonomous driving, and other fields. However, the limited scale of labelled 3D outdoor datasets has posed significant challenges in training an effective model. Manually annotating 3D outdoor datasets is expensive and time-consuming. For example, annotating one scene in the large-scale SemanticKITTI [1] dataset takes around one hour. The whole annotation task on SemanticKITTI [1] requires more than 1700 hours for around 40,000 point cloud frames. However, as the sampling rate of LIDAR used in SemanticKITTI and KITTI [2] is 10Hz, the whole dataset only covers around one hour of real-world data. To address this data annotation challenge, efforts such as self-supervised pre-training [3]–[7], unsupervised learning [8], weakly supervised learning [9], [10], and active learning [11]–[13] have been made. For a detailed survey on label-efficient 3D deep learning for 3D point clouds, please refer to [14]. Our focus in this paper is on weakly supervised learning that only requires sparsely annotated points for outdoor point cloud sequence segmentation.

H. Shi, Z. Wu and G. Lin are with Nanyang Technological University. E-mail: {hanyu001, zhonghua001}@e.ntu.edu.sg, gslin@ntu.edu.sg

F. Liu is with Institute for Infocomm Research A*STAR, Singapore. E-mail: fayaoliu@gmail.com

Y. Xu is with GoerTek Electronics Inc. E-mail: Yi.xu.purdue@gmail.com

Corresponding authors: Guosheng Lin and Fayao Liu

Manuscript received July 26, 2023; revised July 26, 2024.

Recently, some weakly supervised methods [15], [16] have been presented to reduce annotation efforts for outdoor 3D point cloud sequence data, also known as 4D point clouds. As 3D point cloud sequences in outdoor datasets show high temporal correlations among frames, in our previous work W4DTS [15], we designed a two-stage pseudo-label generation approach by leveraging the temporal and spatial correlations. Specifically, in the first stage, W4DTS generates sparsely labelled seeding points based on temporal matching over neighboring point cloud sequences. In the second stage, W4DTS performs label propagation to generate pseudo labels. We achieved impressive results in various settings. However, the decoupled two-stage design has some weaknesses. Since W4DTS relies on a low amount of seeding points from the first stage, even a single false seeding point can lead to significantly inferior quality of the pseudo labels. This reduces the overall quality of the pseudo labels and degrades the segmentation performance of the final model.

To address this issue caused by the two-stage design in W4DTS, this paper proposes a novel Progressive 4D Grouping (**P4G**) approach for generating more reliable pseudo labels. We follow the 0.001% initial annotation setting of W4DTS. Differing from the two-stage design of temporal matching and label propagation in W4DTS, **P4G** builds 4D groups to represent pseudo label clusters and progressively expands these groups with the most similar neighboring points across both time and space. The affinity of neighboring points is

usually robust to low-quality features, and the progressive 4D group expansion ensures the quality of those neighboring points added into existing 4D groups each time. Moreover, to reduce the effect of false pseudo labels, **P4G** refines the affinity score between each pair of points with the affinities between these two points and all the neighboring points over space and time. Using the refined affinity scores, progressive 4D group expansion implicitly utilizes the affinities between unannotated points and the points in existing 4D groups to select unannotated points as new pseudo labels, which is robust to false pseudo labels. In addition, we further design a cross-frame contrastive learning scheme and a local consistency learning scheme to enhance the affinity weights and thus further improve the quality of the 4D groupings.

In **P4G**, we set each initial annotation as a single 4D group. We then design two group expansion strategies to expand the existing groups over space and time. In the two group expansion strategies, we generate a set of weights based on the affinities between points and their neighboring points. We then search for points with high weights to the points in the groups in terms of both time and space and progressively add more points into relevant 4D groups. Points in each 4D group are expected to have high confidence in belonging to the same class. We assign the labels of the initial annotations in the 4D groups to all other points in the same group. These points are the generated pseudo labels used in our model training stage.

We then design a cross-frame contrastive learning scheme to train an affinity enhancement module (a GNN network) for improving the quality of 4D grouping. The cross-frame contrastive learning enlarges the affinity weights of points from the same class. Our experiments demonstrate that our affinity enhancement module using cross-frame contrastive learning can help improve the precision and quantity of pseudo labels from our 4D grouping. Moreover, we adopt a local consistency learning scheme to improve our model and our 4D grouping with unannotated points.

We evaluate our framework on SemanticKITTI [1], SemanticPOSS [17] and ScribbleKITTI [9]. The comparison results are shown in Fig. 1, which manifest the effectiveness of our self-training approach for weakly supervised 3D semantic segmentation.

We summarize the main contributions as follows:

- We design a progressive 4D grouping method to generate pseudo labels for weakly supervised semantic segmentation. Our method progressively expands existing groups containing the initial annotations in 4D space. With this 4D grouping approach, we generate high-quality pseudo labels to train the segmentation network.
- We design two group expansion strategies to expand existing groups over space and time. With the usage of affinities between neighboring points, our group expansion strategies are robust to false pseudo labels, which commonly exist in outdoor point clouds.
- We adopt cross-frame contrastive learning and local consistency learning to enhance the affinity weights in our 4D grouping, thereby improving the quality of the pseudo labels.
- We evaluate our proposed method with 0.001% initial annotations on SemanticKITTI and SemanticPOSS. Our experimental results show that our approach outperforms W4DTS by 4.3% mIoU on the test set of SemanticKITTI. On SemanticPOSS, our model achieves an mIoU score of 55.8%, which is comparable to the fully supervised baseline. On ScribbleKITTI, our method achieves an mIoU score of 53.7% using 0.08% annotations and 56.7% using 0.4% annotations, significantly outperforming competing methods.

II. Related Work

We first provide a general survey on label-efficient 3D deep learning for 3D point clouds, and then focus on the most related area of weakly supervised point cloud segmentation.

A. Label-efficient 3D Deep Learning for 3D Point Clouds

This is an active research area focusing on reducing the dependency on large amounts of labeled data for training deep learning models on 3D point clouds. For a comprehensive survey on this topic, please refer to [14].

Self-supervised learning leverages intrinsic structures and patterns within the unlabeled data to create supervisory signals for learning robust feature representations, which can then be transferred to downstream tasks in a label-efficient manner. Early attempts focus on designing pre-text tasks, such as auto-encoding [18], occlusion completion [19], jigsaw puzzle solving [20] etc. Later on, as a popular self-supervised learning technique, contrastive learning has been extensively explored for 3D point cloud data [3]–[5], [7], [21]. In [7], the authors propose a method by leveraging temporal consistencies among Lidar scans within a contrastive framework. As another line of effort, active learning aims to tackle the 3D data annotation challenge by actively selecting more informative data for labelling within a limited annotation budget. Works such as [11], [13] propose different labelling strategies by considering various uncertainty and diversity measures. Shi et al. [12] investigate the labelling sample granularity for active selection under a limited annotation budget. GrowSP [8] presents the first unsupervised learning method for 3D point clouds. There are also some works that leverage external knowledge to address label-efficient 3D deep learning. For instance, recent efforts leverage advances in vision or vision-language foundation models for 3D point clouds [6], [22]–[24]. These methods typically utilize 2D-3D correspondences to distill the knowledge learned in various foundation models such as CLIP [25], SAM [26], X-Decoder [27] for 3D point cloud segmentation. As a means to directly reduce labeling costs, weakly supervised methods rely on incomplete or inexact supervision, e.g., partial labeling [28], [29], scene-level annotations [30], subcloud level annotations [31], sparse point annotations [32]–[35], scribble annotations [9]. We provide more discussions on weakly supervised point cloud segmentation below.

B. Weakly Supervised Point Cloud Segmentation

Weakly supervised point cloud segmentation is attracting increasing attention. Earlier efforts such as MPRM [31] and GPFN [36] make use of 2D image segmentation to support weakly training of 3D tasks. Xu et al. [34], Zhang et al. [33] and OTOC [32] only annotate a low proportion of points in the point clouds and generate more pseudo labels based on their self-training mechanisms. SQN [37] interpolates the features of unannotated points for the model training. In the outdoor scenario, W4DTS [15] proposes a challenging setting: using 0.1% points of the first frames in each 100-frame point cloud sub-sequence to train an effective model. The key challenge of this setting is to utilize the extremely few initial annotations and those point clouds without any annotation. To tackle this challenge, W4DTS [15] designs a label propagation framework to propagate the initial annotation information to those unannotated frames over both space and time. With the information propagation over both space and time, the label propagation framework selects those points with high confidence scores as the new pseudo labels for model training. LESS [16] proposes a novel weak annotation method and a training framework, which achieves competitive performance compared to the fully supervised counterpart with 0.1% annotations. COARSE3D [38] annotates 0.1% points in every frame and trains the model with a prototype training framework. However, for LESS and COARSE3D, annotating every frame in a large-scale outdoor point cloud dataset is still expensive and may not be necessary as consecutive frames usually contain large redundant information. HybridCR [39] introduces a contrastive learning structure to dig into the consistency locally and globally and achieve improvement on both indoor and outdoor scenes. MulPro [40] generates multiple prototypes with sparse initial annotations and predicts the classes of points based on the affinity between points and those prototypes. Their work focuses on indoor 3D scene segmentation. In [9], the authors propose ScribbleKITTI, the first scribble-annotated Lidar point cloud dataset, along with a self-training pipeline within a student-teacher framework. Compared to [9], our method requires less annotation effort and fully leverages temporal and spatial correlations.

III. Methodology

A. Overview

In the outdoor scenario, LiDAR sensors on a moving car scan the environment to acquire outdoor 3D data sequentially. In KITTI [2], consecutive point cloud frames in the same 3D sequence share a high amount of common regions and objects. Based on this observation, we propose a weakly supervised setting in W4DTS [15], which firstly divides the 3D point cloud sequence into several 100-frame sub-sequences and then only annotates 0.1% points in the first frame of each sub-sequence. This leads to a total annotation budget of 0.001% to enable effective model training. In such a weakly learning setting, there are extremely few initial annotations and a large number of unannotated frames, which limits the performance of models.

To train an effective model, W4DTS initially generates more annotations with an unsupervised super-voxel segmentation [41]. Furthermore, W4DTS designs a two-stage approach using temporal matching and label propagation to propagate the annotations to the whole dataset and generate more pseudo labels. In the first stage, temporal matching selects the most similar unannotated points in the consecutive frame to the annotated points. Those matching results are the new pseudo labels called seed points in W4DTS. With seed points, huge amounts of unannotated frames can be utilized in the model training stage. However, the performance of the final model is still limited by the low amount of seed points. Then, we design the second stage using label propagation to generate more pseudo labels based on the seed points in each frame. Label propagation builds a directed graph in each frame and propagates the prediction information from seed points to unannotated points for refining the prediction scores of unannotated points. With the refined prediction score of unannotated points, we select points with high prediction scores as the new pseudo labels.

However, the quality of pseudo labels in W4DTS is limited by the two-stage design of temporal matching and spatial graph propagation. Temporal matching is an one-to-one matching, which selects a single most similar point to each of the annotated points and existing pseudo labeled points in the previous frame. When temporal matching makes false matches in unannotated frames, these false matches are propagated to later unannotated frames. In such a fashion, errors are easily accumulated in the pseudo labels during temporal matching. Furthermore, label propagation also propagates the information of false seed points in each frame, which generates more false pseudo labels. As temporal matching and spatial graph propagation rely on a low amount of seed points, these false matches significantly reduce the quality of pseudo labels. This limits the overall quality of pseudo labels and degrades the segmentation performance of the final model.

To tackle this issue, we propose a progressive 4D grouping approach to group the extremely sparse initial annotations and unannotated points as 4D groups for generating pseudo labels. As the affinity of neighboring points is usually robust to noise, progressive 4D grouping gradually expands existing 4D groups with unannotated neighboring points which have high affinities to the points in existing 4D groups. Furthermore, during 4D grouping expansion, we refine the affinity score between each pair of points with the affinities between all the points over space and time. Using this method, we implicitly utilize the affinities between target points and existing points in 4D groups to expand the groups, which are robust to false pseudo labels. Specifically, we set each initial annotated point as a single initial group and progressively expand them with neighboring points that have the highest refined affinity scores to those points in each group over time and space. Each point within a group is assigned a pseudo label by considering the initial annotation in the group. We also design an affinity enhancement module trained with a cross-frame contrastive loss to generate more reliable context information and enlarge the affinity weights of the neighboring points from the same class. Furthermore, we adopt local consistency

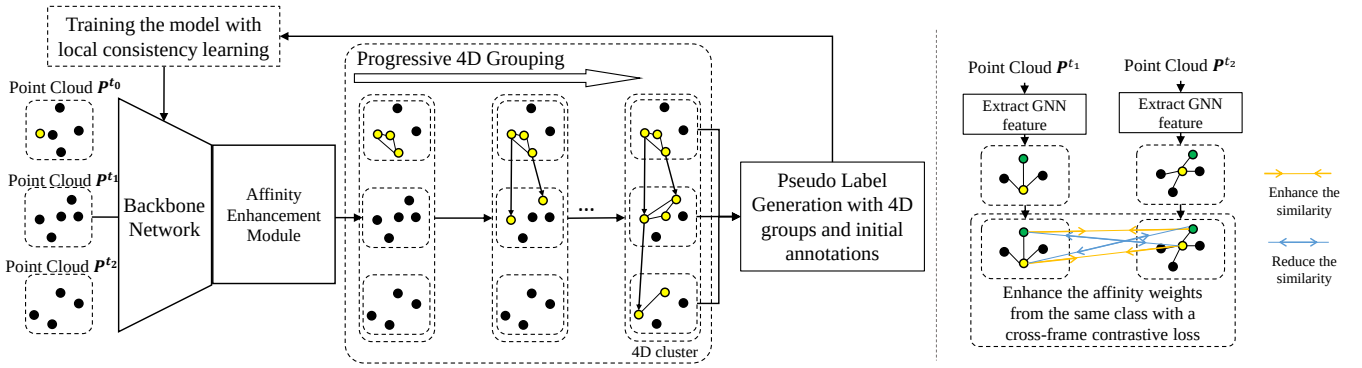


Fig. 2: **An overview of our method.** Firstly, we propose a progressive 4D grouping approach to generate pseudo labels. Secondly, a graph neural network is trained by a cross-frame contrastive loss to further enhance the affinities between neighboring points, and thus improve the 4D grouping performance. Thirdly, we adopt a local consistency learning scheme to leverage unannotated points for our model training.

training to utilize the information of unannotated points during the backbone model training.

A summary of the differences between this extension and W4DTS is shown below:

1) We propose a progressive 4D grouping method to generate high-quality pseudo labels, resolving the issues of the two-stage design of W4DTS. Differing from the one-to-one temporal matching and dense graph propagation approach in W4DTS, **P4G** builds 4D groups using the initial annotations and progressively expands these groups with the most similar neighboring points over space and time. In such a fashion, **P4G** enhances the quality and quantity of pseudo labels compared to those generated by W4DTS. With more high-quality pseudo labels, the training pipeline is simplified and is more efficient than the two-stage training of W4DTS.
2) We conduct additional experiments on ScribbleKitti [9]. The results show that our method significantly outperforms the competing methods.

B. Progressive 4D Grouping

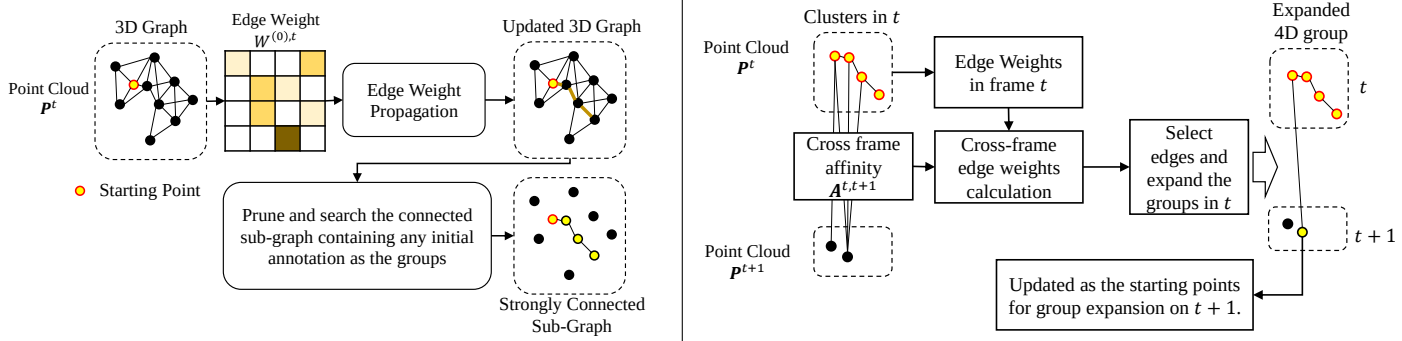
Recent 3D panoptic and instance segmentation works [28], [42]–[44] show that unsupervised or semi-supervised clustering approaches, e.g. HDBSCAN [45], Affinity Propagation [46] are capable of grouping points from the same instance with high quality. This clustering approach groups points based on the feature affinity of neighboring points. The affinity of neighboring points is usually more robust than the affinity between distant points. Furthermore, with a higher affinity, points and their nearby neighboring points have a higher possibility to be in the same class. Considering these assumptions, we propose a progressive 4D grouping approach to group the annotated and unannotated points in 4D space.

Specifically, we build the initial groups with the initial annotations and progressively expand the groups with highly confident points in the 4D space. Each 4D group represents a 4D region where all points within it have high probabilities of belonging to the same class. In our implementation, the progressive 4D grouping approach has two group expansion strategies. For the spatial dimension, we propose a spatial group expansion module to build 3D graphs on the point clouds and use an edge weight propagation approach to increase the weights between points that are strongly connected. With pruning on the 3D graph, this spatial group expansion generates a set of connected sub-graphs containing annotated points as groups. For the temporal dimension, we propose a temporal group expansion module by designing a context-aware edge weighting scheme to generate the edge weights between points in the current frame and points in existing groups from the previous frame. Our context-aware edge weighting is based on the affinity between two nodes and the edge weights of neighboring nodes in the previous frame. With an edge weight selection approach, we expand the groups in the previous frame to the points in the current frame. Then, we set these points in the current frame as starting points in spatial group expansion. After we progressively build the 4D groups, the labels of points within each 4D group are assigned the labels of initial annotations in each group. Eventually, we use these points as pseudo labels for training our segmentation model. We show an overview of our proposed approach in Fig. 2.

Group Expansion over Space. Initially, we follow W4DTS [15] and OTOC [32] to apply super-voxel segmentation as a pre-processing step on all the point clouds. We use average pooling over the points in each super-voxel. The feature $\bar{\mathbf{f}}_i^t$, coordinate $\bar{\mathbf{c}}_i^t$, and prediction probability score $\bar{\mathbf{y}}_i^t$ of the i-th super-voxel in the t-th frame $\mathbf{P}^t$ are:

$$\begin{aligned} \bar{\mathbf{c}}_i^t &= \frac{1}{n}\sum_{l=1}^{n}\mathbf{c}_l^t; \\ \bar{\mathbf{f}}_i^t &= \frac{1}{n}\sum_{l=1}^{n}\mathbf{f}_l^t; \\ \bar{\mathbf{y}}_i^t &= \frac{1}{n}\sum_{l=1}^{n}\mathbf{y}_l^t, \end{aligned} \tag{1}$$

where n is the number of points in the i-th super-voxel. Firstly,



(a) Group Expansion over Space. (b) Group Expansion across Time.

Fig. 3: **Two modules in our progressive 4D grouping approach**. In Figure (a), the spatial group expansion module generates 4D groups based on the adjacency of nodes in the graph. We search for connected sub-graphs from the starting points and expand existing groups with points connected to any points in the groups. In Figure (b), the temporal group expansion module generates cross-frame edge weights between the points in frames t and $t+1$. Then, we select the most confident edges and assign the nodes in $t+1$ connected to those edges to existing relevant groups. These points in groups are the starting points in the spatial group expansion module for frame $t+1$.

we build a radius graph on point cloud $\mathbf{P}^t$, with nodes being super-voxels and edges connecting neighboring super-voxels. In our implementation, the maximum radius for constructing edges between pairs of super-voxels is $1.2m$. The initial edge weights $\mathbf{W}^{(0),t}$ of the 3D graph are the original affinity matrix, with its (i,j)-th entry being $w_{i,j}^{(0),t}$.

We first define the feature distance $d_{i,j}^{f,t}$, probability score distance $d_{i,j}^{y,t}$ and coordinate distance $d_{i,j}^{c,t}$ between super-voxels i and j as:

$$\begin{aligned} d_{i,j}^{f,t} &= 1 - \frac{\bar{\mathbf{f}}_i^t \cdot \bar{\mathbf{f}}_j^t}{||\bar{\mathbf{f}}_i^t|| ||\bar{\mathbf{f}}_j^t||}; \\ d_{i,j}^{y,t} &= 1 - \frac{\bar{\mathbf{y}}_i^t \cdot \bar{\mathbf{y}}_j^t}{||\bar{\mathbf{y}}_i^t|| ||\bar{\mathbf{y}}_j^t||}; \\ d_{i,j}^{c,t} &= 1 - \exp\left(-\frac{||\bar{\mathbf{c}}_i^t - \bar{\mathbf{c}}_j^t||^2}{2}\right). \end{aligned} \tag{2}$$

We then calculate the initial affinity weight $w_{i,j}^{(0),t}$ as:

$$w_{i,j}^{(0),t} = \mathbb{1}_{i,j}^t \cdot \exp\left(-\frac{d_{i,j}^{f,t}}{\tau^f} - \frac{d_{i,j}^{y,t}}{\tau^y} - \frac{d_{i,j}^{c,t}}{\tau^c}\right), \tag{3}$$

where τ^f, τ^y and τ^c are hyper-parameters to weight different distance metrics. $\mathbb{1}_{i,j}^t$ indicates the adjacency of super-voxel i and j, which equals 1 if there is an edge between the two super-voxels and 0 otherwise.

To achieve high-quality grouping, we design an edge weight propagation scheme to increase the edge weights between nodes that are strongly connected and reduce the edge weights between nodes that have weak connections. To perform edge weight propagation, we define a transition matrix $\mathbb{T}^{(k),t}$ on the graph, with its (i,j)-th entry calculated as $w_{i,j}^{(k),t} / \sum_l w_{i,l}^{(k),t}$. Here l indexes all the super-voxels in the frame $\mathbf{P}^t$. The iterative edge weight propagation can be formulated as:

$$\mathbf{W}^{(k+1),t} = \alpha \mathbf{W}^{(k),t} + (1-\alpha) \cdot \mathbb{T}^{(k),t} \mathbf{W}^{(k),t}. \tag{4}$$

Here k represents the k-th iteration. α is an updating ratio hyper-parameter to balance the updating weights and the initial values. Moreover, we observe that the final model only improves minorly when the number of iterations is larger than 3. Therefore, we set the maximum iteration number as 3 in our implementation.

Next, the updated edge weights are used to calculate the final adjacency matrix and search for connected sub-graphs to expand the groups. Specifically, we remove those unreliable edges according to the updated edge weight matrix $\mathbf{W}^{(k),t}$ and build the final adjacency matrix $\mathbf{A}^t$ as:

$$a_{i,j}^t = \begin{cases} 1 & \text{if } w_{i,j}^{(k),t} \geqslant \lambda^{spatial} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Note that $\lambda^{spatial}$ is a threshold hyper-parameter for edge pruning, and $w_{i,j}^{(k),t}$ is an edge weight in $\mathbf{W}^{(k),t}$. A Breadth-First Search (**BFS**) is then adopted to search over the adjacency matrix $\mathbf{A}^t$ for sub-graphs containing the initial annotations. We use the initial annotations as the starting points for **BFS**. Note that for those frames without any initial annotations, we use the results of temporal group expansion as the starting points. With **BFS**, all the super-voxels connected to the query nodes or search starting nodes are added to the connected sub-graph of initial annotations. Ultimately, we expand the existing groups with the super-voxels in the connected sub-graphs.

Group Expansion across Time. If a point cloud $\mathbf{P}^{t+1}$ ($t \geqslant 1$) is not annotated, the starting point of the spatial group expansion is missing. To generate pseudo initial annotations for the spatial group expansion in the current frame $\mathbf{P}^{t+1}$, we propose a temporal group expansion module by following a similar design as the spatial group expansion. Specifically, we build a graph across two succeeding frames $\mathbf{P}^{t+1}$ and $\mathbf{P}^t$. The nodes are the super-voxels assigned to all the super-voxels in $\mathbf{P}^{t+1}$ and any group in point cloud $\mathbf{P}^t$. The edges connect the nodes in $\mathbf{P}^{t+1}$ and $\mathbf{P}^t$. This is a bipartite graph, and also a radius graph. Note that the coordinates of $\mathbf{P}^{t+1}$ and

Algorithm 1 Pseudo Label Generation

Input: Point clouds $\mathbf{P}^0, ..., \mathbf{P}^t$, Initial annotations $\mathbf{L}^0$
Output : Pseudo labels $\mathbf{S}^0, ..., \mathbf{S}^t$

procedure
  # $\mathbf{G}^l$ and $\tilde{\mathbf{G}}^l$ are the 4D graphs for 4D grouping.
  $\tilde{\mathbf{P}}^{-1}, \tilde{\mathbf{L}}^{-1}, \tilde{\mathbf{G}}^{-1} \leftarrow RemoveUnannotatedPoints(\mathbf{P}^0, \mathbf{L}^0)$
  for $l = 0, ..., t$ **do**
    # $\tilde{\mathbf{L}}^l$ are pseudo initial annotations in point cloud $\mathbf{P}^l$.
    $\mathbf{L}^l, \mathbf{G}^l \leftarrow GroupExpansionAcrossTime(\tilde{\mathbf{P}}^{l-1}, \tilde{\mathbf{L}}^{l-1}, \tilde{\mathbf{G}}^{l-1}, \mathbf{P}^l)$
    # $\mathbf{S}^l$ are the generated pseudo labels in $\mathbf{P}^l$.
    $\mathbf{S}^l, \mathbf{G}^l \leftarrow GroupExpansionOverSpace(\mathbf{P}^l, \mathbf{L}^l, \mathbf{G}^l)$
    $\tilde{\mathbf{P}}^l, \tilde{\mathbf{L}}^l, \tilde{\mathbf{G}}^l \leftarrow RemoveUnannotatedPoints(\mathbf{P}^l, \mathbf{S}^l, \mathbf{G}^l)$
  end for
end procedure

$\mathbf{P}^t$ are aligned based on the position information, and all the radii in graph building are based on the aligned coordinates. In our implementation, the maximum radius is set to $1.2m$ for background points (e.g. road and building), and $5m$ for object points (e.g. car and person). The affinity score $w_{j,i}^{(0),t+1,t}$ of super-voxels p_j^{t+1} and p_i^t are calculated similarly as per Eq. (3).

Next, we design a context-aware edge weight updating scheme based on not only the affinity between two nodes in each edge but also the information from other nodes in the previous frame. Specifically, we obtain the updated edge weight $w_{j,i}^{t+1,t}$ by aggregating the edge weights between p_j^{t+1} and all the neighbouring super-voxels in the previous frame. We first define a cross-frame weight matrix $\mathbf{D}^{t+1,t}$, with its (j,i)-th entry $\delta_{j,i}^{t+1,t}$ calculated as $w_{j,i}^{(0),t+1,t} / \sum_l w_{j,l}^{(0),t+1,t}$. Here l indexes all the super-voxels in the frame $\mathbf{P}^t$. The context-aware edge weight updating is formulated as

$$v_{j,i}^{t+1,t} = \sum_{l \in \mathcal{N}} \delta_{j,l}^{t+1,t} \cdot w_{l,i}^{(k),t}. \tag{6}$$

Note that $\mathcal{N}$ denotes the set of all row indexes in $\mathbf{W}^{(k),t}$, i.e., all the super-voxels in frame $\mathbf{P}^t$. The matrix formulation of the updated edge weights $\mathbf{W}^{t+1,t}$ is

$$\mathbf{V}^{t+1,t} = \mathbf{D}^{t+1,t} \cdot \mathbf{W}^{(k),t}. \tag{7}$$

Subsequently, we search for the most reliable connections between the super-voxel p_j^{t+1} and the groups in the previous frame. For each super-voxel p_j^{t+1}, we select edges with the maximum edge weight $v_{j,i}^{t+1,t}$. We further remove all the edges whose $v_{j,i}^{t+1,t} < \lambda^{temporal}$. Here $\lambda^{temporal}$ is a threshold for edge pruning on the temporal graph. If there is still an edge between p_j^{t+1} and a super-voxel p_i^t in the previous frame, we expand the 4D group of p_i^t with p_j^{t+1}. The super-voxel p_j^{t+1} are assigned as the starting points for spatial group expansion in frame $\mathbf{P}^{t+1}$.

Pseudo Label Generation. We show the pseudo code of our pseudo label generation pipeline in Algorithm 1. $\mathbf{G}^l$ are the graphs generated by our progressive 4D grouping approach, and $\tilde{\mathbf{G}}^l$ are the graphs with the unannotated points removed from $\mathbf{G}^l$. Note that $GroupExpansionAcrossTime$ and $GroupExpansionOverSpace$ are the **Group Expansion across Time** and the **Group Expansion over Space** techniques we presented earlier. The function $RemoveUnannotatedPoints$ indicates that we remove those points that are not in any group.

As mentioned in W4DTS [15], the amount of object points only consists of 0.1% of the points in the whole dataset. The extreme data unbalance issue limits the quality of pseudo labels and the performance of the final model. Therefore, we only consider the points of the object classes (e.g. car, person and bike) in $SpatialGroupExpanding$. For background classes (e.g. road, building, and tree), we directly use the results from $TemporalGroupExpanding$ to update the pseudo labels. Furthermore, in the first iteration of pseudo label generation, for each background point p_i^t, $TemporalGroupExpanding$ only keeps the edges with the maximum value in $\{w_{i,l}^{t,t+1}, l \in \mathcal{M}\}$ where $\mathcal{M}$ is the index set of all the points in frame $t+1$.

C. Affinity Enhancement Module

To further improve our 4D grouping results, we design an affinity enhancement module and use a contrastive loss to generate enhanced feature affinity weights. Firstly, we build a k nearest neighbor graph at the super-voxel level and build a GNN network to extract more graph context information. Similar to the Grid-GCN [47], we extract the super-voxel features by using multi-layer perceptrons (MLPs) and aggregating the features based on the relative positions of neighboring nodes. Specifically, we first use an MLP to adapt the input features (which is a concatenation of the feature $\bar{\mathbf{f}}_i^t$ and the prediction score $\bar{\mathbf{y}}_i^t$) of super-voxel p_i^t:

$$\mathbf{s}_i^t = MLP([\bar{\mathbf{f}}_i^t, \bar{\mathbf{y}}_i^t]). \tag{8}$$

In our graph neural network layer, we use the features of neighboring nodes and the feature differences between the target node and its neighboring nodes as the input. We then determine the edge weights of the neighboring nodes with the class of the target node and the relative position of neighboring nodes. Our graph neural network layer is formulated as:

$$\begin{aligned} \mathbf{e}_i^t = \mathbf{s}_i^t + \sum_{\mu} & Tanh(MLP([\bar{\mathbf{c}}_i^t - \bar{\mathbf{c}}_\mu^t, \bar{\mathbf{y}}_i^t])) \\ & \times ReLU(MLP([\mathbf{s}_\mu^t, \mathbf{s}_i^t - \mathbf{s}_\mu^t])). \end{aligned} \tag{9}$$

Here $\mathbf{e}_i^t$ denotes the enhanced features of super-voxel p_i^t and μ indexes its neighbours. The two MLPs have the same output dimension. Here we use a $Tanh$ function to generate the edge weights. Then, we assemble two graph neural network layers for our final output $\mathbf{e}_i^t$. In our implementation, the dimensions of $\mathbf{s}_i^t$ and $\mathbf{e}_i^t$ are 256 and the number of classess respectively.

Then, we design a cross-frame contrastive loss to optimize the GNN parameters by enforcing features from the same class to be similar. After each training epoch of the backbone network, we randomly sample pairs of annotated frames from the same sequence. For each annotated super-voxel in two frames t_1 and t_2, we calculate the feature $\mathbf{e}_i^{t_1}$ from the graph neural network. For a super-voxel $p_i^{t_1}$, we sample one positive example and several negative examples in the t_2 frame. The class of the positive sample is the same as $p_i^{t_1}$ while the classes

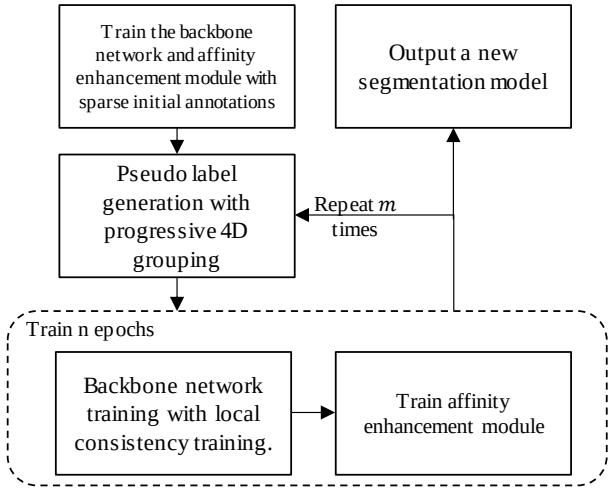


Fig. 4: **An overview of our training framework**.

	$\mathbf{S}_1$	$\mathbf{S}_2$
τ^f_{space}	0.6	0.6
τ^e_{space}	0.6	0.6
τ^y_{space}	0.6	0.6
τ^c_{space}	0.6	0.6
τ^f_{time}	0.045	0.045
τ^e_{time}	0.045	0.045
τ^y_{time}	0.045	0.045
τ^c_{time}	0.045	0.045
λ_{space}	0.6	0.8
λ_{time}	0.6	0.8
α	0.5	0.5

TABLE I: **The hyper-parameter settings in our final model**. τ^f_{space},τ^e_{space},τ^y_{space} and τ^c_{space} are the hyper-parameters in spatial group expansion. τ^f_{time},τ^e_{time},τ^y_{time} and τ^c_{time} are the hyper-parameters in temporal group expansion.Similarly, λ_{space} and λ_{time} are the thresholds in group expansion in space and time respectively. $\mathbf{S}_1$ and $\mathbf{S}_2$ are the first and second pseudo label generation iterations in our framework.

of negative samples are different. We apply the InfoNCE loss [48] on the super-voxel level features as below:

$$l^{enhance} = \sum_{l} -\log \frac{\exp\left(\mathbf{e}_l^{t_1} \cdot \mathbf{e}_j^{+,t_2}\right)}{\sum_k \exp\left(\mathbf{e}_l^{t_1} \cdot \mathbf{e}_k^{t_2}\right) + \exp\left(\mathbf{e}_l^{t_1} \cdot \mathbf{e}_j^{+,t_2}\right)}, \tag{10}$$

where l indexes super-voxels in $\mathbf{P}^{t_1}$ and k indexes negative samples. $\mathbf{e}_j^{+,t_2}$ is the positive sample. Afterwards, we add one MLP after the affinity enhancement module to predict super-voxel level prediction scores. The final loss is $l^{total} = l^{seg,sv} + l^{enhance}$ where $l^{seg,sv}$ is a cross-entropy loss at the super-voxel level. In the training stage, we train the backbone network and our affinity enhancement module separately.

The distance of two GNN features is calculated as:

$$d_{i,j}^{e,t_1,t_2} = 1 - \frac{\mathbf{e}_i^{t_1} \cdot \mathbf{e}_j^{t_2}}{\|\mathbf{e}_i^{t_1}\| \|\mathbf{e}_j^{t_2}\|}. \tag{11}$$

For two points in the same class, the distance of two GNN features is:

$$d_{i,j}^{e,t} = 1 - \frac{\mathbf{e}_i^{t} \cdot \mathbf{e}_j^{t}}{\|\mathbf{e}_i^{t}\| \|\mathbf{e}_j^{t}\|}. \tag{12}$$

Then the enhanced affinity score $w_{i,j}^t$ as per Eq. (3) is:

$$w_{i,j}^t = \mathbb{1}_{i,j}^t \cdot \exp(-\frac{d_{i,j}^{f,t}}{\tau^f} - \frac{d_{i,j}^{y,t}}{\tau^y} - \frac{d_{i,j}^{c,t}}{\tau^c} - \frac{d_{i,j}^{e,t}}{\tau^s}). \tag{13}$$

τ^s is a weighting hyper-parameter. Similarly, the enhanced affinity score $w_{i,j}^{t,t+1}$ for two nodes of the graph in group expansion across time is:

$$w_{i,j}^{t,t+1} = \mathbb{1}_{i,j}^{t,t+1} \cdot \exp(-\frac{d_{i,j}^{f,t,t+1}}{\tau^f} - \frac{d_{i,j}^{y,t,t+1}}{\tau^y} - \frac{d_{i,j}^{c,t,t+1}}{\tau^c} - \frac{d_{i,j}^{e,t,t+1}}{\tau^s}). \tag{14}$$

Note $d_{i,j}^{f,t,t+1}$, $d_{i,j}^{y,t,t+1}$, $d_{i,j}^{c,t,t+1}$ and $d_{i,j}^{e,t,t+1}$ are the distances of features, prediction scores, coordinates and the enhanced features between points from frame t and $t+1$.

D. Local Consistency Learning

In the backbone training stage, we adopt a local consistency training scheme to leverage unannotated points. The local consistency training also enhances the affinity of points from the same classes and thus improves the quality of 4D group expansion. Specifically, we add noise to each point in a point cloud and enforce the predictions of the original and deformed points to be similar. Firstly, we apply a random augmentation to the input point clouds. The augmentation operation includes random rotation, random mirroring, and a random perturbation to each point. For a single point, the coordinate of its perturbed version $\mathbf{c}_i^{aug,t}$ is $\{x_i^t + \xi^x, y_i^t + \xi^y, z_i^t + \xi^z\}$, where ξ^x, ξ^y and ξ^z are sampled from a Gaussian distribution $\mathcal{N}(0, 0.025^2)$. Then, we enforce the predictions $\mathbf{y}_i^t$ and $\mathbf{y}_i^{aug,t}$ to be similar by optimizing an L2 loss between them. This loss is termed as l^{local}. Then, we combine the cross-entropy loss l^{seg} and the local consistency loss l^{local} to train the backbone network. As the local consistency loss aims to leverage unlabelled data to improve model training, it becomes less effective and incurs high computation costs when there is sufficient data with pseudo labels. Therefore, we only apply the local consistency loss in the first iteration.

E. Training Details

We show an overview of our weakly supervised training pipeline in Fig. 4. Note that m is the iteration number of pseudo label generations, which is set as 2 in our experiments. Our backbone network is the same as our previous work, which is MinkUnet [49]. We initially pretrain the backbone model and affinity enhancement module with sparse initial annotations. Here the sparse initial annotations are also the same as our previous work. In our previous work, we uniformly sampled 1 point for each object (e.g. car, person) and 20 points for the environment objects (e.g. road, building). Then, we iteratively update the pseudo labels and fine-tune the backbone model. In each iteration, we first update the pseudo labels with progressive 4D grouping. With the new pseudo labels, we train the backbone model and the affinity enhancement module in order. After finishing the m-th iteration, we use the backbone network as the final segmentation model.

We also show a detailed setting of hyper-parameters in Tab. I. Note that we have two pseudo label generation iterations, which we define as $\mathbf{S}_1$ and $\mathbf{S}_2$. We set low values of parameters τ^f_{time}, τ^e_{time}, τ^y_{time} and τ^c_{time} to enlarge the difference between each point and thus only expand the high confidence points into the existing 4D group across time. With this setting, we significantly reduce the error accumulation over time. In our experiment, the hyper-parameter settings can be migrated to different datasets and annotation settings. We achieve the best performance with the same hyper-parameter settings on all the experiments shown in Section IV.

During the model training, our device is a workstation with an Intel i9-9900X CPU, an Nvidia RTX 3090, and 16GB memory. The whole training process is around 3.5 days. Compared to the training of the fully supervised backbone model, the additional computational costs are around 1 day. Specifically, the training of the affinity enhancement module and the pseudo label updating cost around 10 hours (6 hours for affinity enhancement and 4 hours for two pseudo label updating iterations). The local consistency training takes around 10 hours of additional training time.

IV. Experimental Results

Following the weakly supervised learning setting in W4DTS [15], we evaluate our proposed method on SemanticKITTI [1], SemanticPOSS [17] and ScribbleKITTI [9] by using the same weak annotation settings.

SemanticKITTI is a LiDAR point cloud dataset of outdoor scenes, which contains 19 annotated sequences with over 40,000 frames. Specifically, there are 19,130 frames for training and 4,071 frames for validation. The rest 20,351 frames are for testing. In the weakly setting of W4DTS, there are only 198 annotated frames in the training set. For each annotated frame, they only use 0.01% of annotated points. The total annotation budget is around 0.0057%.

SemanticPOSS is also an outdoor LiDAR point cloud dataset. There are 2988 frames in 6 sequences. In SemanticPOSS, W4DTS uniformly samples 26 frames and annotates an average of 3000 points in each frame, which covers around 5% points. The total annotation budget is around 0.05%.

ScribbleKITTI reannotates the same point clouds dataset as SemanticKITTI, which only covers 8.06% annotations of SemanticKITTI. Similar to our validation settings, we use 4,071 fully annotated frames to evaluate our methods.

A. Results on SemanticKITTI

Comparison on the test set of SemanticKITTI. The results on the test set of SemanticKITTI are shown in Tab. II. Note that we also show some results of fully supervised approaches. With the 0.001% initial annotations, our framework outperforms the baseline method (MinkUnet) with an absolute mIoU boost of 9.7%. Our approach also achieves a 4.1% improvement compared to W4DTS [15]. SQN [37] and COARSE3D [38] are 3D-based approaches and require annotations on every frame. Our approach achieves 3.3% and 2.9% boosts over SQN using 0.1% annotations and COARSE3D using 0.01% annotations respectively. Another weak setting is to annotate 0.1% points in the first frame of a 20-frame sub-sequence, which leads to an overall annotation budget of 0.005%. On this 0.005% weak annotation setting, the performance of our framework is 1.8% higher than W4DTS, and only shows a 1.6% margin compared to COARSE3D using 0.1% annotations. *It is worth mentioning that our method is comparable to the fully supervised baseline MinkUnet (54.1 vs. 56.2).*

	Backbone	Supervision	mIoU
DarkNet21Seg [1]	-	100%	47.4
RandLA-Net [50]	-	100%	53.9
MinkUnet [49]	-	100%	56.2
KPconv [51]	-	100%	58.8
SalsaNext [52]	-	100%	59.5
SPVCNN [53]	-	100%	66.4
Cylinder3D [54]	-	100%	67.8
$(AF)^2$-S3Net [55]	-	100%	69.7
PVKD [56]	-	100%	71.2
PTv3 [57]	-	100%	75.5
SQN [37]	RandLA-Net [50]	0.1%	50.8
SQN [37]	RandLA-Net [50]	0.01%	39.1
COARSE3D [38]	SalsaNext [52]	0.1%	**55.7**
COARSE3D [38]	SalsaNext [52]	0.01%	46.2
MinkUnet [49]	-	0.001%	39.4
W4DTS-100f [15]	MinkUnet [49]	0.001%	44.8
Ours-100f	MinkUnet [49]	0.001%	**49.1**
MinkUnet [49]	-	0.005%	46.4
W4DTS-20f [15]	MinkUnet [49]	0.005%	52.3
Ours-20f	MinkUnet [49]	0.005%	**54.1**

TABLE II: **Results on the test set of SemanticKITTI**. We implement our method with two settings in W4DTS on SemanticKITTI. Ours-100f and Ours-20f denote the annotated frames sampled every 100 and every 20 frames respectively.

	SGE	TGE	AE	LC	LC†	mIoU
MinkUnet						60.7
Baseline						40.9
W4DTS-GM [15]						49.2
W4DTS-OT [15]						50.3
Model-A	√					50.7
Model-B	√	√				51.5
Model-C	√	√	√			52.9
Model-D	√	√	√		√	54.0
Model-E	√	√	√	√		**54.6**

TABLE III: **Ablation study on the validation set of SemanticKITTI**. MinkUnet is our fully-supervised backbone network. SGE is the spatial group expansion, and TGE is the temporal group expansion. AE is the affinity enhancement module. LC is the local consistency training scheme, and LC† denotes that we train the model with local consistency training in all iterations. Model-A: we only use SGE and replace TGE with the greedy matching in W4DTS. We compare our method with W4DTS using greedy matching (W4DTS-GM) and W4DTS using optimal transport (W4DTS-OT).

Ablation study on the validation set of SemanticKITTI. We compare different variants of our proposed method against the baseline and W4DTS on the validation set of SemanticKITTI in Tab. III. Our backbone network is the same as W4DTS, which is a 42-layer MinkUnet [49]. Here MinkUnet denotes the fully supervised version of MinkUnet, and Baseline is

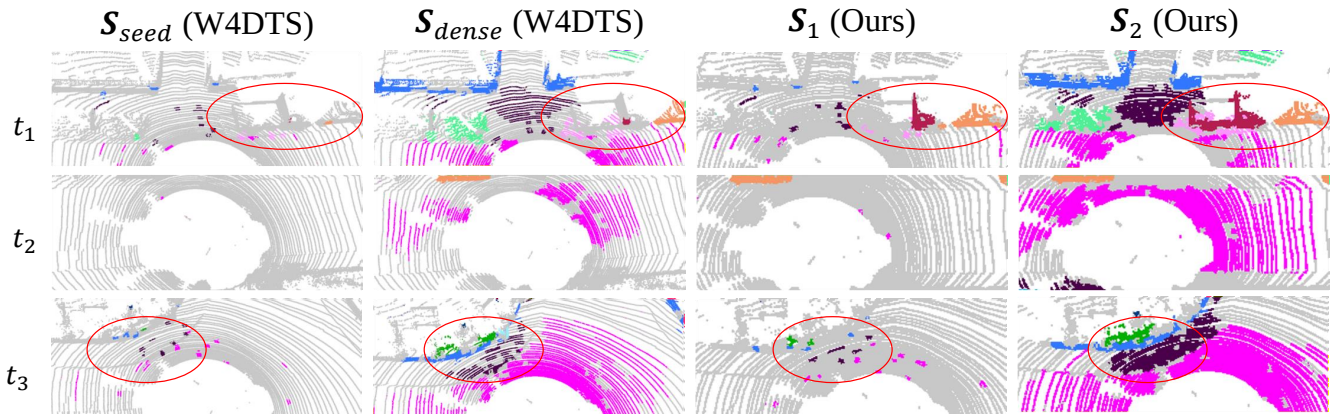


Fig. 5: **Qualitative comparison of pseudo labels generated by W4DTS [15] and our method**. Here $\mathbf{S}_{seed}$ and $\mathbf{S}_{dense}$ denote the seed propagation and dense propagation stages in W4DTS. $\mathbf{S}_1$ and $\mathbf{S}_2$ are the first and second pseudo label generation iterations in our framework.

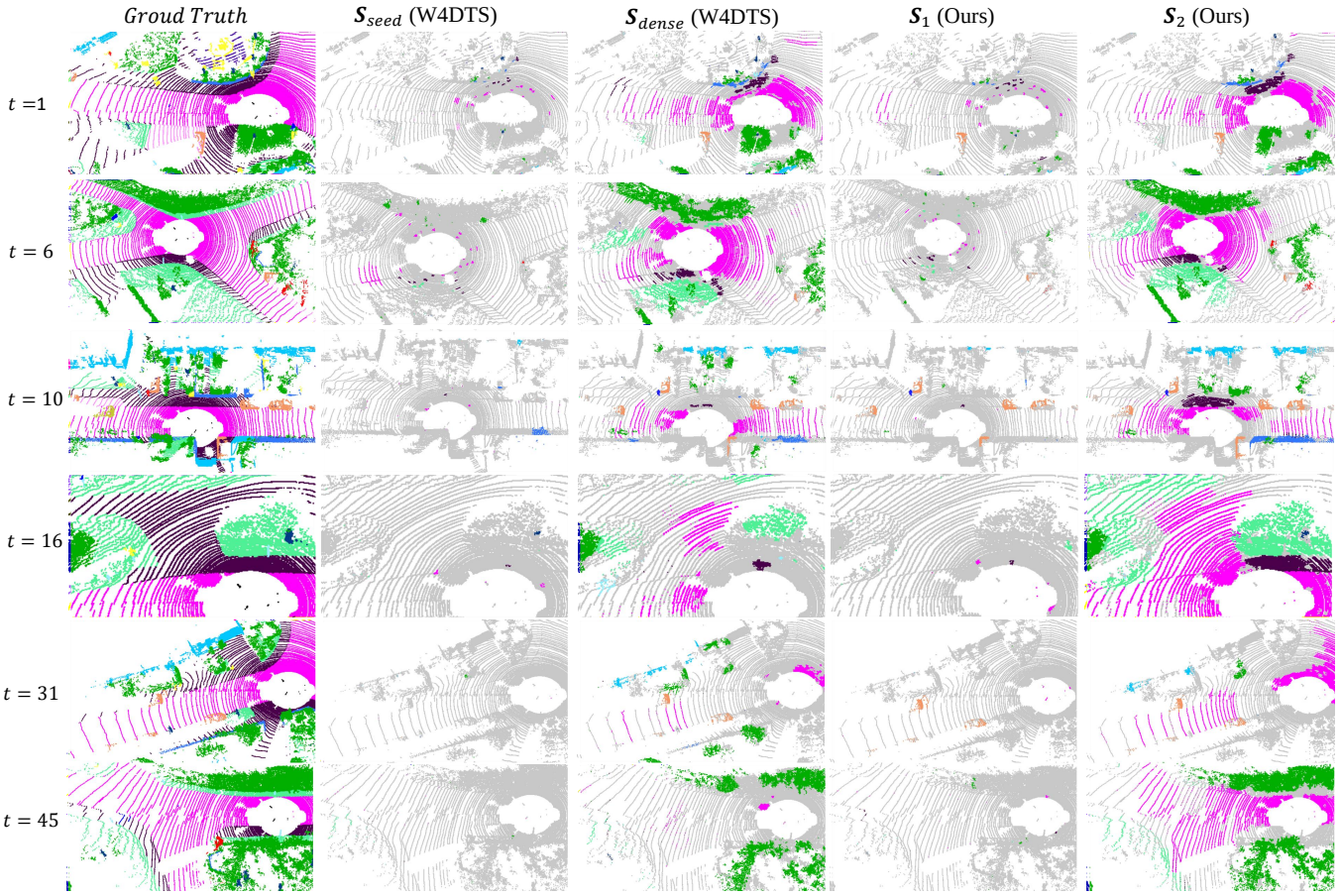


Fig. 6: **Qualitative comparison of pseudo labels generated by W4DTS and our method in different frames**. Here $\mathbf{S}_{seed}$ and $\mathbf{S}_{dense}$ denote the seed propagation and dense propagation stages in W4DTS. $\mathbf{S}_1$ and $\mathbf{S}_2$ are the first and second pseudo label generation iterations in our framework. t is the timestamp of the frame in each point cloud sub-sequence.

the weakly supervised MinkUnet using the initial annotation naively. If we only use the spatial group expansion and replace the temporal group expansion module with the greedy matching method in W4DTS (Model-A), the mIoU score is 0.4% higher than W4DTS-OT and 1.5% higher than W4DTS-GM. Our method with only the progressive 4D grouping approach (Model-B) achieves an improvement of 1.2% over W4DTS-OT and 2.3% over W4DTS-GM. Note that W4DTS-OT solves an optimal transport problem to generate one-to-one matching during temporal matching, which requires

	$\mathbf{S}_1$	$\mathbf{S}_2$
Model-A	48.4	50.7
Model-B	48.9	51.5
Model-C	49.1	52.9
Model-D	50.8	54.0
Model-E	50.8	54.6

TABLE IV: **Ablation study of our modules in two pseudo label updating iterations on SemanticKITTI**. Here $\mathbf{S}_1$ is the first iteration of pseudo label generation, and $\mathbf{S}_2$ is the second iteration.

	$\mathbf{S}_1$		$\mathbf{S}_2$	
	mIoU	Sparseness	mIoU	Sparseness
Initial	73.1	0.0057%		
W4DTS [15]	78.1	0.8%	56.6	19.8%
Model-A	73.6	3.0%	76.7	3.2%
Model-B	74.7	3.2%	66.7	38.6%
Model-C	78.1	3.7%	72.8	42.8%
Model-D	78.1	3.7%	74.0	40.8%
Model-E	78.1	3.7%	74.0	40.8%

TABLE V: **Ablation study on the pseudo labels of SemanticKITTI**. For updated pseudo labels, we use two metrics, i.e. mIoU scores on updated pseudo labels and the sparseness of updated pseudo labels over the whole dataset, to evaluate the quality of pseudo labels. Note that we use the percentage of pseudo labels in the whole dataset to represent the sparseness of pseudo labels. Here we only show the quality of pseudo labels which achieve the best performance of relevant models. Similarly, we also show the quality of pseudo labels of W4DTS [15]. For W4DTS, S_1 represents seed propagation stage, i.e. S_{seed}, and S_2 represents dense propagation stage, i.e. S_{dense}.

much higher computation costs than Model-A and Model-B. This demonstrates the effectiveness of our progressive 4D grouping approach. Compared to the greedy matching, our temporal group expansion module (Model-B) shows a 0.8% improvement over Model-A. Further equipped With the affinity enhancement module (Model-C), our method achieves an improvement of 1.4% over Model-B. Compared to Model-C, our full model with the local consistency training brings an improvement of 1.7% in terms of mIoU. As mentioned in Section III-E, we only use the local consistency training in the first iteration. If we apply local consistency training in all the iterations (Model-D), it only shows a 1.1% improvement over Model-C. Notably, *the performance of our full model (Model-E) increases the mIoU score by 13.7% over the baseline.*

Ablation study of our modules in two pseudo label generation iterations on SemanticKITTI We show an ablation study of each module in two pseudo label generation iterations on SemanticKITTI in Tab. IV. Note that all the models are the same as the models in the ablation study on the validation set of SemanticKITTI. Model-D and Model-E share the same model in the first iteration S_1. Group expansion across time (Model B) improves Model-A (+0.5% in S_1 and +0.8% in S_2). Comparing Model-B and Model-C, we can see that when

	Backbone	Supervision	mIoU
PointNet++ [58]	-	100%	20.1
SalsaNext [52]	-	100%	45.0
RandLA-Net [59]	-	100%	53.5
KPConv [51]	-	100%	55.2
MinkUnet [49]	-	100%	56.3
JS3C-Net [60]	-	100%	60.2
COARSE3D [38]	SalsaNext [52]	0.1%	43.0
COARSE3D [38]	SalsaNext [52]	0.01%	31.1
MinkUnet [49]	-	0.05%	39.5
W4DTS [15]	MinkUnet [49]	0.05%	52.2
Ours	MinkUnet [49]	0.05%	**55.8**

TABLE VI: **Results on the data part 3 of SemanticPOSS.** Our method achieves very close performance compared to the fully supervised backbone network MinkUnet.

the amount of pseudo labels is low in S_1, the performances of Model-B (48.9% of mIoU) and Model-C (49.1%) are similar. However, when the amount of pseudo labels increases in S_2, the affinity enhancement module achieves an absolute mIoU boost of 1.4% (52.9 vs. 51.5). Furthermore, with the local consistency training scheme, Model-E achieves a 1.7% improvement over M ccodel-C in S_1 and S_2 respectively (50.8 vs. 49.1, 54.6 vs. 52.9). This shows the effectiveness of the local consistency training scheme.

Ablation study on the pseudo labels of SemanticKITTI To investigate the effectiveness of our framework, we further compare the mIoU scores and the sparseness of our pseudo labels. We show the results in Tab. V. Note that the mIoU scores and the sparseness can not entirely determine the final performance of the model. Therefore, we only use these two metrics as references for model training and hyper-parameter tuning. Using progressively 4D grouping, the pseudo label quality of our Model-B outperforms that from W4DTS in terms of mIoU score and sparseness in both $\mathbf{S}_1$ and $\mathbf{S}_2$. Compared to Model-B, our affinity enhancement module (Model-C) significantly improve the mIoU scores and sparseness of pseudo labels in both S_1 and S_2. As Model-C and Model-E share the same model and affinity enhancement module trained by the sparse initial annotations, the pseudo labels in S_1 are the same. With the local consistency training, the pseudo labels of Model-E in S_2 are 1.2% higher than the pseudo labels of Model-C in S_2. Due to the fact that the model performance of Model-E is 1.7% higher than Model-C, the quality of pseudo labels is actually improved with local consistency training even though the pseudo labels in Model-E in S_2 is more sparse than the pseudo labels of Model-C in S_2. Model-D and Model-E share the same model for both pseudo label generation iterations, and thus the pseudo labels for Model-D and Model-E are the same.

Qualitative results of our pseudo labels We show a qualitative comparison of the generated pseudo labels between W4DTS and our method in Fig. 5. In all three frames, the precision and quantity of our pseudo labels are significantly better than that of W4DTS. With our progressive 4D grouping approach, the pseudo labels show a high regional consistency. Especially for the object classes, our approach generates dense pseudo labels with high quality. As highlighted by red

	Backbone	Supervision	mIoU
MinkUnet [49]	-	100%	61.1
SPVCNN [53]	-	100%	63.8
Cylinder3D [54]	-	100%	64.3
ScribbleKITTI [9]	MinkUnet [49]	8%	58.5
ScribbleKITTI [9]	SPVCNN [53]	8%	60.8
ScribbleKITTI [9]	Cylinder3D [54]	8%	61.3
MeanTeacher [61]	-	4%	53.9
CutMix-Seg [62]	-	4%	54.3
CBST [63]	-	4%	54.6
CPS [64]	-	4%	54.8
LaserMix [65]	Cylinder3D [54]	4%	**58.7**
MeanTeacher [61]	-	1.6%	52.8
CutMix-Seg [62]	-	1.6%	52.9
CBST [63]	-	1.6%	53.3
CPS [64]	-	1.6%	53.9
LaserMix [65]	Cylinder3D [54]	1.6%	55.6
MeanTeacher [61]	-	0.8%	50.1
CutMix-Seg [62]	-	0.8%	50.7
CBST [63]	-	0.8%	50.7
CPS [64]	-	0.8%	51.8
LaserMix [65]	Cylinder3D [54]	0.8%	53.7
MeanTeacher [61]	-	0.08%	41.0
CutMix-Seg [62]	-	0.08%	36.7
CBST [63]	-	0.08%	41.5
CPS [64]	-	0.08%	41.4
LaserMix [65]	Cylinder3D [54]	0.08%	44.2
Ours	MinkUnet [49]	0.4%	**56.7**
Ours	MinkUnet [49]	0.08%	**53.7**

TABLE VII: **Results on the test set of ScribbleKITTI**. We present the results of our models using one annotated frame every 100 frames and 20 frames, which aligns with our settings on SemanticKITTI. Note that the supervision ratios pertain to the entire SemanticKITTI dataset. As ScribbleKITTI annotates 8% points of SemanticKITTI, the settings of 0.08%, 0.4%, 0.8%, 1.6%, and 4% supervisions correspond to using one scribble annotated frame every 100 frames, 20 frames, 10 frames, 5 frames, and 2 frames respectively in our setup.

circles in the first row of Fig. 5, our method generates more comprehensive pseudo labels of object classes even when we use a model trained only by sparse initial annotations in stage S_1. For background objects, the pseudo labels in the red circles in frame t_3 also show the effectiveness of our progressive 4D grouping.

Furthermore, in Fig. 6, we also show a qualitative comparison of the ground truths, the generated pseudo labels from W4DTS and our method in different point cloud frames. We randomly sample the frames at different timestamps in different sub-sequences. Similar to the observation in W4DTS, the quality and quantity of pseudo labels in the frames far from the first frame are lower than those in the frames close to the first frame. In all the frames, our pseudo labels have higher precision and quantity than that of W4DTS.

B. Results on SemanticPOSS

Next, we evaluate our method on the SemanticPOSS. The results are shown in Tab. VI. Note the result of SalsaNext [52] is reported by COARSE3D [38]. With the same backbone network, our method achieves an absolute mIoU boost of 3.6% over W4DTS. Furthermore, the mIoU score of our method is 12.8% higher than COARSE3D [38]. *Notably, our method achieves very close performance compared to the fully supervised backbone network MinkUnet (only by a margin of 0.5%).*

C. Results on ScribbleKITTI

We also evaluate our method on ScribbleKITTI [9], with results shown in Tab. VII. We present results obtained using two different settings: 0.08% and 0.4% annotations. These settings align with those we used in SemanticKITTI. As ScribbleKITTI annotates 8% points of SemanticKITTI, the settings of 0.08% and 0.4% annotations correspond to using scribble annotations of one frame every 100 frames and every 20 frames, respectively, in our setup. Note that the performances of MeanTeacher [61], CutMix-Seg [62], CBST [63] and CPS [64] using 0.08% and 1.6% supervision settings are reported by LaserMix [65]. *When using the same 0.08% annotations, our model significantly outperforms LaserMix [65] by a margin of 9.5%.* Notably, the backbone network of LaserMix [65] is Cylinder3D [54], which outperforms our backbone network MinkUnet. Furthermore, the performance of our model using 0.4% annotations is comparable to LaserMix that uses 10 times our annotations (56.7 vs. 58.7).

V. Conclusion

We propose a weakly supervised segmentation framework to train effective models with 0.001% annotations on outdoor 3D datasets. Specifically, we design a progressive 4D grouping approach to associate unannotated and annotated points for generating high-quality pseudo labels.We propose two group expansion strategies to group points in 4D space based on their affinities, i.e. spatial group expansion and temporal group expansion. We further adopt cross-frame contrastive learning and local consistency learning to improve the quality of pseudo labels and thus our final performance. Experimental results show our method achieves significant improvements over previous best works.

Acknowledgment This research is supported by the Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021).

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9297–9307.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[3] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *European Conference on Computer Vision*, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12348. Springer, 2020, pp. 574–591.

[4] L. Nunes, R. Marcuzzi, X. Chen, J. Behley, and C. Stachniss, "Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination," *IEEE Robotics Autom. Lett.*, vol. 7, no. 2, pp. 2116–2123, 2022.

[5] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3d features on any point-cloud," in *IEEE/CVF International Conference on Computer Vision*. IEEE, 2021, pp. 10 232–10 243.

[6] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu, "Segment any point cloud sequences by distilling vision foundation models," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.

[7] L. Nunes, L. Wiesmann, R. Marcuzzi, X. Chen, J. Behley, and C. Stachniss, "Temporal consistent 3d lidar representation learning for semantic perception in autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2023, pp. 5217–5228.

[8] Z. Zhang, B. Yang, B. Wang, and B. Li, "Growsp: Unsupervised semantic segmentation of 3d point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2023, pp. 17 619–17 629.

[9] O. Unal, D. Dai, and L. Van Gool, "Scribble-supervised lidar semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2697–2707.

[10] Y. Su, X. Xu, and K. Jia, "Weakly supervised 3d point cloud segmentation via multi-prototype learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7723–7736, 2023. [Online]. Available: https://doi.org/10.1109/TCSVT.2023.3281151

[11] X. Shi, X. Xu, K. Chen, L. Cai, C. S. Foo, and K. Jia, "Label-efficient point cloud semantic segmentation: An active learning approach," *CoRR*, vol. abs/2101.06931, 2021. [Online]. Available: https://arxiv.org/abs/2101.06931

[12] T. Wu, Y. Liu, Y. Huang, H. Lee, H. Su, P. Huang, and W. H. Hsu, "Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation," in *2021 IEEE/CVF International Conference on Computer Vision*. IEEE, 2021, pp. 15 490–15 499.

[13] B. Xie, S. Li, Q. Guo, C. H. Liu, and X. Cheng, "Annotator: A generic active learning baseline for lidar semantic segmentation," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.

[14] A. Xiao, X. Zhang, L. Shao, and S. Lu, "A survey of label-efficient deep learning for 3d point clouds," *CoRR*, vol. abs/2305.19812, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.19812

[15] H. Shi, J. Wei, R. Li, F. Liu, and G. Lin, "Weakly supervised segmentation on outdoor 4d point clouds with temporal matching and spatial graph propagation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 830–11 839.

[16] M. Liu, Y. Zhou, C. R. Qi, B. Gong, H. Su, and D. Anguelov, "Less: Label-efficient semantic segmentation for lidar point clouds," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, 2022, pp. 70–89.

[17] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "Semanticposs: A point cloud dataset with large quantity of dynamic instances," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 687–693.

[18] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. J. Guibas, "Learning representations and generative models for 3d point clouds," in *International Conference on Learning Representations*. OpenReview.net, 2018.

[19] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *IEEE/CVF International Conference on Computer Vision*. IEEE, 2021, pp. 9762–9772.

[20] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 942–12 952.

[21] F. Liu, G. Lin, C. Foo, C. K. Joshi, and J. Lin, "Point discriminative learning for data-efficient 3d point cloud analysis," in *International Conference on 3D Vision*. IEEE, 2022, pp. 42–51.

[22] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li, "Pointclip: Point cloud understanding by CLIP," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 8542–8552.

[23] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, and W. Wang, "Clip2scene: Towards label-efficient 3d scene understanding by CLIP," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2023, pp. 7020–7030.

[24] S. Dong, F. Liu, and G. Lin, "Leveraging large-scale pretrained vision foundation models for label-efficient 3d point cloud segmentation," *CoRR*, vol. abs/2311.01989, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2311.01989

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 8748–8763.

[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár, and R. B. Girshick, "Segment anything," in *IEEE/CVF International Conference on Computer Vision*. IEEE, 2023, pp. 3992–4003.

[27] X. Zou, Z. Dou, J. Yang, Z. Gan, L. Li, C. Li, X. Dai, H. Behl, J. Wang, L. Yuan, N. Peng, L. Wang, Y. J. Lee, and J. Gao, "Generalized decoding for pixel, image, and language," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2023, pp. 15 116–15 127.

[28] Y. Liao, H. Zhu, Y. Zhang, C. Ye, T. Chen, and J. Fan, "Point cloud instance segmentation with semi-supervised bounding-box mining," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[29] L. Kong, J. Ren, L. Pan, and Z. Liu, "Lasermix for semi-supervised lidar semantic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2023, pp. 21 706–21 716.

[30] C. Yang, M. Chen, Y. Chuang, and Y. Lin, "2d-3d interlaced transformer for point cloud segmentation with scene-level supervision," in *IEEE/CVF International Conference on Computer Vision*. IEEE, 2023, pp. 977–987.

[31] J. Wei, G. Lin, K.-H. Yap, T.-Y. Hung, and L. Xie, "Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4384–4393.

[32] Z. Liu, X. Qi, and C.-W. Fu, "One thing one click: A self-training approach for weakly supervised 3d semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1726–1736.

[33] Y. Zhang, Z. Li, Y. Xie, Y. Qu, C. Li, and T. Mei, "Weakly supervised semantic segmentation for large-scale point cloud," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3421–3429.

[34] X. Xu and G. H. Lee, "Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 706–13 715.

[35] Y. Su, X. Xu, and K. Jia, "Weakly supervised 3d point cloud segmentation via multi-prototype learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7723–7736, 2023.

[36] H. Wang, X. Rong, L. Yang, J. Feng, J. Xiao, and Y. Tian, "Weakly supervised semantic segmentation in 3d graph-structured point clouds of wild scenes," *arXiv preprint arXiv:2004.12498*, 2020.

[37] Q. Hu, B. Yang, G. Fang, Y. Guo, A. Leonardis, N. Trigoni, and A. Markham, "Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*. Springer, 2022, pp. 600–619.

[38] R. Li, A.-Q. Cao, and R. de Charette, "Coarse3d: Class-prototypes for contrastive learning in weakly-supervised 3d point cloud segmentation," 2022.

[39] M. Li, Y. Xie, Y. Shen, B. Ke, R. Qiao, B. Ren, S. Lin, and L. Ma, "Hybridcr: Weakly-supervised 3d point cloud semantic segmentation via hybrid contrastive regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 930–14 939.

[40] Y. Su, X. Xu, and K. Jia, "Weakly supervised 3d point cloud segmentation via multi-prototype learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[41] Y. Lin, C. Wang, D. Zhai, W. Li, and J. Li, "Toward better boundary preserved supervoxel segmentation for 3d point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 143, pp. 39 – 47, 2018, iSPRS Journal of Photogrammetry and Remote Sensing Theme Issue "Point Cloud Processing". [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271618301370

[42] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu, "Lidar-based panoptic segmentation via dynamic shifting network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 090–13 099.

[43] R. Razani, R. Cheng, E. Li, E. Taghavi, Y. Ren, and L. Bingbing, "Gp-s3net: Graph-based panoptic sparse semantic segmentation network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 076–16 085.

[44] L. Tang, L. Hui, and J. Xie, "Learning inter-superpoint affinity for weakly supervised 3d instance segmentation," vol. 13841, pp. 176–192, 2022.

[45] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection,"

ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 10, no. 1, pp. 1–51, 2015.

[46] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[47] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann, "Grid-gcn for fast and scalable point cloud learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5661–5670.

[48] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[49] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[50] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Learning semantic segmentation of large-scale point clouds with random sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8338–8354, 2021.

[51] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," pp. 6410–6419, 2019.

[52] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*. Springer, 2020, pp. 207–222.

[53] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European Conference on Computer Vision*. Springer, 2020, pp. 685–702.

[54] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, "Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation," *arXiv preprint arXiv:2008.01550*, 2020.

[55] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 547–12 556.

[56] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8479–8488.

[57] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler faster stronger," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4840–4851.

[58] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017.

[59] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.

[60] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, "Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion," pp. 3101–3109, 2021.

[61] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[62] G. French, S. Laine, T. Aila, M. Mackiewicz, and G. Finlayson, "Semi-supervised semantic segmentation needs strong, varied perturbations," *arXiv preprint arXiv:1906.01916*, 2019.

[63] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 289–305.

[64] X. Chen, Y. Yuan, G. Zeng, and J. Wang, "Semi-supervised semantic segmentation with cross pseudo supervision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2613–2622.

[65] L. Kong, J. Ren, L. Pan, and Z. Liu, "Lasermix for semi-supervised lidar semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 705–21 715.