Expert Systems With Applications 306 (2026) 130824

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

You only click once: single point weakly supervised 3D instance segmentation for autonomous driving

Guangfeng Jiang [a], Jun Liu [a,*], Yongxuan Lv [a], Yuzhi Wu [a], Xianfei Li [b], Wenlong Liao [b], Tao He [b], Pai Peng [b,*]

[a] Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, 230027, China
[b] COWAROBOT, Shanghai, China

ARTICLE INFO

Keywords:
LiDAR point clouds
Autonomous driving
3D instance segmentation
Weak supervision

ABSTRACT

Outdoor LiDAR point cloud 3D instance segmentation is a crucial task in autonomous driving. However, it requires laborious human efforts to annotate the point cloud for training a segmentation model. To address this challenge, we propose **YoCo**, a framework that generates 3D pseudo labels using minimal coarse click annotations in the bird's eye view plane. It is a significant challenge to produce high-quality pseudo labels from sparse annotations. Our YoCo framework first leverages vision foundation models combined with geometric constraints from point clouds to enhance pseudo label generation. Second, a temporal and spatial-based label updating module is designed to generate reliable updated labels. It leverages predictions from adjacent frames and utilizes the inherent density variation of point clouds (dense near, sparse far). Finally, to further improve label quality, an IoU-guided enhancement module is proposed, replacing pseudo labels with high-confidence and high-IoU predictions. Experiments on the Waymo dataset demonstrate YoCo's effectiveness and generality, achieving state-of-the-art performance among weakly supervised methods and surpassing fully supervised Cylinder3D. Additionally, the YoCo is suitable for various networks, achieving performance comparable to fully supervised methods with minimal fine-tuning using only 0.8% of the fully labeled data, significantly reducing annotation costs.

1. Introduction

Autonomous driving has attracted significant attention from both academia and industry due to its potential to transform transportation systems (Tsitsokas et al., 2023). This technology encompasses a range of critical tasks, including perception (such as object detection and segmentation) and control systems. A major challenge in autonomous driving is to improve safety and economic efficiency while optimizing decision-making processes in complex traffic environments (Ma et al., 2025; Wang et al., 2025).

One of the key research areas in autonomous driving is 3D point cloud segmentation, which plays a fundamental role in tasks like obstacle detection and environmental mapping. However, the current state-of-the-art segmentation models rely heavily on dense point-wise annotations, which are both labor-intensive and costly. For example, annotating a single scene in the ScanNet dataset (Dai et al., 2017) takes an average of 22.3 minutes. Reducing the reliance on such dense annotations, through weakly supervised or semi-supervised approaches, is a critical challenge that can significantly lower data collection costs and improve the scalability of autonomous driving systems.

Recent advancements in neural network architectures (Wu et al., 2024; Yan et al., 2018; Zhao et al., 2021; Zhu et al., 2021) and the emergence of high-quality autonomous driving datasets (Behley et al., 2019; Caesar et al., 2020; Geiger et al., 2013) have driven progress in segmentation techniques. However, despite these developments, minimizing annotation costs remains a crucial area of research to further enhance the feasibility and efficiency of autonomous vehicle systems.

Recent studies have attempted to address the problem of weakly supervised segmentation on 3D point clouds. Existing methods utilize various types of weak labels, such as sparse point-level labels (Chen et al., 2024; Hu et al., 2022; Zhang et al., 2022), scribble-level labels (Unal et al., 2022; Wang et al., 2024), and box-level labels (Jiang et al., 2024; Li et al., 2024; Ngo et al., 2023; Yu et al., 2024). However, most of these approaches focus on semantic segmentation, while instance segmentation is more complex, as it requires distinguishing different instances within the same semantic category. For 3D instance segmentation tasks,

* Corresponding authors.
E-mail addresses: junliu@ustc.edu.cn (J. Liu), pengpai@cowarobot.com (P. Peng).

https://doi.org/10.1016/j.eswa.2025.130824
Received 21 September 2025; Received in revised form 25 November 2025; Accepted 12 December 2025
Available online 25 December 2025
0957-4174/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

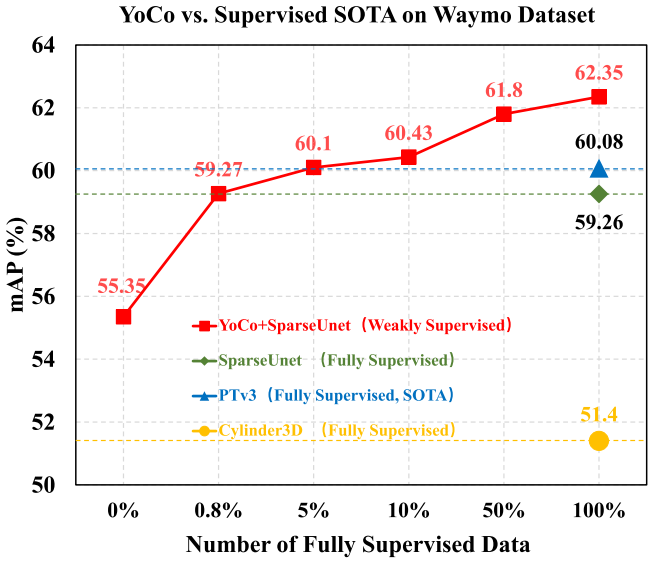


Fig. 1. 3D instance segmentation performance comparison. The weakly supervised YoCo for fine-tuning compared to fully supervised methods. The results show that our YoCo outperforms fully supervised Cylinder3D without fine-tuning (0%). Fine-tuning YoCo with 0.8% and 5% labeled frames data exceeds the fully supervised SparseUnet and state-of-the-art (SOTA) PTv3, respectively. The dashed line at 100% denotes the performance of the model trained under full supervision using all annotated samples in the training dataset.

the annotation of 3D bounding boxes is still expensive, although methods (Li et al., 2024; Ngo et al., 2023; Yu et al., 2024) using 3D bounding boxes as weak supervision have achieved promising results. A recent work (Jiang et al., 2024) explores weakly supervised instance segmentation for LiDAR point clouds using low-cost 2D bounding boxes as weak labels, but this approach still shows a noticeable performance gap compared to fully supervised methods.

Inspired by the work mentioned above, we rethink whether a method with lower labeling cost can achieve better instance segmentation performance and even narrow the gap between weakly and fully supervised approaches.

With this motivation, we propose a single-point supervised instance segmentation framework called **YoCo**. To reduce annotation cost while preserving instance-level precision, YoCo requires only a single click annotation per object in the bird's eye view (BEV) plane. Each click serves as a weak supervision cue that can be transformed into a dense 3D pseudo label through a vision-geometry pipeline. Specifically, the clicked point is projected into the corresponding camera view and used as a prompt for SAM (Kirillov et al., 2023) to generate a 2D mask, following the idea in MWSIS (Jiang et al., 2024). The resulting mask is then re-projected back into the 3D space to form an initial pseudo label. This step-by-step process effectively bridges sparse single-click supervision and dense 3D instance annotations. However, a major challenge lies in the limited zero-shot capability of SAM, resulting in noisy or inaccurate 2D masks. Therefore, a critical challenge lies in filtering high-quality 3D pseudo labels from these noisy outputs.

To address this challenge, we introduce a 3D pseudo label generation module based on the vision foundation models (VFMs), named VFM-PLG. Specifically, we use click annotations to obtain the corresponding 2D masks through VFMs, then leverage the geometric constraints of the corresponding 3D masks to filter out high-quality 3D pseudo labels. In addition, to further improve the quality of pseudo labels, we take advantage of the generalization and robustness of neural networks by introducing two key modules: the temporal and spatial-based label updating (TSU) module, and the intersection-over-union (IoU)-guided label enhancement (ILE) module. The TSU module refines and updates the pseudo labels by incorporating predictions from adjacent frames, while the ILE module further enhances label quality by replacing lower-quality labels offline with more accurate predictions.

Experimental results show that our YoCo significantly outperforms previous SOTA weakly supervised 3D instance segmentation methods and even surpasses fully supervised Cylinder3D (Zhu et al., 2021), as shown in Fig. 1. Additionally, YoCo demonstrates strong generality, making it suitable for various networks. Moreover, by fine-tuning the model with only 0.8% of fully supervised data, it can surpass the performance of the fully supervised baseline.

In summary, our contributions are summarized as follows: (1) To the best of our knowledge, we are the first to propose using click annotations for instance segmentation of outdoor LiDAR point clouds. This method significantly reduces the burden of instance segmentation annotations. (2) We propose the VFM-PLG, which combines VFMs with geometric constraints information of the object to generate high-quality pseudo labels. In addition, the TSU and ILE modules further improve pseudo label quality by leveraging the generalization and robustness of neural networks. (3) Extensive experiments on the Waymo dataset demonstrate that our YoCo achieves SOTA performance in weakly supervised instance segmentation, surpassing fully supervised methods such as Cylinder3D. Furthermore, YoCo exhibits strong generality, making it applicable across various networks.

2. Related work

2.1. LiDAR-based fully supervised 3D segmentation.

Existing 3D LiDAR point cloud segmentation methods can be divided into three types according to data representation: point-based, projection-based, and voxel-based.

Point-based methods (Cheng et al., 2024; Qi et al., 2017a,b; Thomas et al., 2019; Wu et al., 2019, 2024, 2022; Zeng et al., 2024; Zhao et al., 2021) directly use raw point clouds as the input. The classic PointNet (Qi et al., 2017a) utilizes the permutation invariance of both point-wise MLPs and pooling layers to aggregate features across a set. KPConv (Thomas et al., 2019) and PointConv (Wu et al., 2019) construct continuous convolution to directly process 3D points. The Point Transformer series (Wu et al., 2024, 2022; Zhao et al., 2021) adopts the transformer architecture to extract features from 3D points. Projection-based methods (Ando et al., 2023; He et al., 2025; Kong et al., 2023; Milioto et al., 2019; Park et al., 2023) project 3D points onto 2D images to form regular representations, allowing the use of well-established neural networks from 2D image processing. RangeViT (Ando et al., 2023) directly applies a pre-trained ViT model as the encoder and fine-tunes it, demonstrating the feasibility of transferring 2D knowledge to 3D tasks. Rangeformer (Kong et al., 2023) and RangeNet++ (Milioto et al., 2019) use an encoder-decoder hourglass-shaped architecture as the backbone for feature extraction. Other methods (Cheng et al., 2021; Choy et al., 2019; Graham et al., 2018; Zhu et al., 2021) convert point clouds into regular 3D voxelization. SSCN (Graham et al., 2018) introduces sparse convolutional networks to handle voxelized sparse point clouds. Cylinder3D (Zhu et al., 2021) introduces 3D cylindrical partitioning and asymmetrical 3D convolutions to handle the sparsity and varying density of outdoor point clouds.

While point-based methods deliver high performance, they come with significant computational costs due to the large-scale raw LiDAR data. On the other hand, projection-based methods are more efficient but lose valuable internal geometric information, resulting in suboptimal performance. Taking both time and memory efficiency into account, we adopt voxelized representations and select a sparse convolutional U-Net (Shi et al., 2020) as our backbone.

2.2. Weakly supervised 3D instance segmentation.

Point cloud segmentation has made significant progress in fully supervised settings. However, dense point-wise annotation is costly. To

reduce the annotation burden, some work (Chen et al., 2024; Guo et al., 2024; Hu et al., 2022; Jiang et al., 2024; Li et al., 2024; Ngo et al., 2023; Unal et al., 2022; Wang et al., 2024; Wei et al., 2020; Yu et al., 2024; Zhang et al., 2022) has explored using various weak supervision signals. For 3D instance segmentation tasks, 3D bounding boxes provide coarse information about instance objects, making instance segmentation feasible. Box2Mask (Li et al., 2024) is the first work to use 3D bounding boxes as weak supervision labels. GaPro (Ngo et al., 2023) proposes a gaussian process method to address pseudo label ambiguity in overlapping regions of multiple 3D bounding boxes. CIP-WPIS (Yu et al., 2024) leverages 2D instance knowledge and 3D geometric constraints to handle the 3D bounding box perturbation issues. Additionally, MWSIS (Jiang et al., 2024) is the first work to use 2D bounding boxes as weak supervision signals for outdoor point cloud segmentation. It introduces various fine-grained pseudo label generation and refinement methods, and explores the possibility of integration with SAM (Kirillov et al., 2023). However, both 2D and 3D bounding boxes still involve considerable annotation costs. Our method only requires a click on the object in the BEV plane to generate high-quality pseudo labels.

2.3. Click-level annotation for 3D perception tasks.

Click-level annotation is a highly efficient and labor-saving labeling method. The recent work (Liu et al., 2023, 2021; Meng et al., 2021, 2020; Tao et al., 2022; Xia et al., 2024; Zhang et al., 2023) has begun to incorporate it into various 3D perception tasks.

One Thing One Click (Liu et al., 2021) employs click-level labels and introduces a graph propagation module to iteratively generate semantic pseudo labels. SegGroup (Tao et al., 2022) propagates click-level labels to unlabeled segments through iterative grouping, generating instance pseudo labels. Meanwhile, ClickSeg (Liu et al., 2023) presents a method that uses k-means clustering with fixed initial seeds to generate instance pseudo labels online. In the field of 3D object detection, WS3Ds (Meng et al., 2021, 2020) annotate object centers in the BEV plane. It utilizes these center clicks as supervision signals to generate cylindrical proposals, and then employs a small amount of ground truth to train the network to produce 3D bounding boxes. ViT-WSS3D (Zhang et al., 2023) proposes using a vision transformer to construct a point-to-box converter.

The aforementioned studies, especially those focusing on weakly supervised 3D instance segmentation, primarily address indoor point clouds. In contrast, 3D instance segmentation of outdoor LiDAR point clouds remains largely unexplored. Although MWSIS (Jiang et al., 2024) utilizes 2D bounding boxes, which bring lower annotation costs, it still exhibits a significant performance gap compared to fully supervised methods. To further minimize annotation costs, we propose a weakly supervised instance segmentation framework that relies solely on click-level annotations, effectively narrowing the gap with fully supervised approaches.

3. Method

Our goal is to generate high-quality 3D instance pseudo labels using sparse click-level annotations, and to narrow the performance gap between weakly supervised and fully supervised methods. To achieve this, we propose a simple yet effective framework, **YoCo**, which integrates pseudo label generation with network training, as illustrated in Fig. 2. By leveraging the minimal input of click annotations, YoCo efficiently creates reliable pseudo labels that maintain strong performance, even with limited supervision. The detailed process is outlined as follows:

For pseudo label generation in Fig. 2(a), given a set of calibrated images and point cloud data, we first annotate the point cloud with click-level labels in the BEV plane. These labels are then projected onto the corresponding images, and processed by our proposed VFM-PLG module. The VFM-PLG leverages the VFMs and geometric constraints from the point cloud to generate high-quality 3D pseudo labels, as described in Section 3.2.

For network training in Fig. 2(b), we adopt the MeanTeacher (Tarvainen & Valpola, 2017) method, which involves a student network and a teacher network. The teacher network is updated using an exponential moving average (EMA) of the student's weights. It predicts labels from adjacent frames, and the TSU module uses these predictions to refine the 3D pseudo labels generated by the VFM-PLG. This refinement incorporates temporal and spatial information from adjacent frames, as detailed in Section 3.3. Additionally, to further boost the reliability of the pseudo labels, we introduce the ILE module. This module enhances the labels offline by using high-confidence and high-IoU results to update the 3D pseudo labels, further improving the performance of our method, as outlined in Section 3.4.

3.1. Preliminary

Given a set of calibrated images and point cloud data, we utilize sensor calibration to project the point cloud onto the images, establishing a mapping relationship between the 3D points and image pixels. Specifically, consider a set of points $\mathbf{P^{3d}} = \{p_i^{3d}\}_{i=1}^{N} = \{(x_i, y_i, z_i)\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$ we can obtain the corresponding pixel coordinates $\mathbf{P^{2d}} = \{p_i^{2d}\}_{i=1}^{N} = \{(u_i, v_i)\}_{i=1}^{N} \in \mathbb{R}^{N\times 2}$ by applying the projection transformation formula:

$$z_c(u_i, v_i, 1)^T = \mathbf{M}_{int} \times \mathbf{M}_{ext} \times (x_i, y_i, z_i, 1)^T \tag{1}$$

where N is the number of the points, z_c represents the depth of the point in the camera coordinate system, $\mathbf{M}_{int}$ and $\mathbf{M}_{ext}$ denote the intrinsic and extrinsic parameters of the camera, respectively.

3.2. VFM-based pseudo label generation

SAM (Kirillov et al., 2023) is a vision foundation model that inputs images and prompts to generate corresponding 2D masks. The prompts include points, bounding boxes, masks, and texts. By leveraging the projection relationship described in Eq. (1), we project the click-level annotations from point clouds onto the image as prompts to obtain the object's 2D masks. Points that fall within these 2D masks can be considered as 3D pseudo labels. This process can be formally expressed by the following equation:

$$m_i = Color(SAM(I_i, p_i, c_i)) \tag{2}$$

where m_i denotes the 3D pseudo label to the i-th click annotation, *Color* represents the operation of mapping a 2D mask to a 3D pseudo label, I_i, p_i, and c_i represent the image, 2D coordinate, and the class label corresponding to the i-th click annotation, respectively.

However, this approach faces three challenges. First, SAM struggles with segmenting composite categories like cyclists. The second one is inaccuracies in the SAM segmentation mask and the projection relationship. Finally, due to the lack of height information in the BEV plane, there may be multiple corresponding 3D points in the current click, and an incorrect prompt will result in an inaccurate segmentation result.

To address the first issue, we utilize the Depth Anything Model (DAM) (Yang et al., 2024) as an auxiliary tool to perform depth-based smoothing, particularly for composite categories like cyclists (as shown in Fig. 3). Specifically, the image is processed through the DAM to generate a depth map, and the depth map and image features are then used to interact with the prompt's features, resulting in the corresponding 2D mask. (as shown in Fig. 4). Therefore, Eq. (2) is updated as:

$$m_i = \begin{cases} Color(SAM(I_i), p_i, c_i), & c_i \text{ is composite categories} \\ Color(SAM(DAM(I_i), p_i, c_i)), & \text{other categories} \end{cases} \tag{3}$$

As for the last two issues, point cloud geometric constraints are proposed for filtering the labels. Specifically, we first obtain a 3D pseudo

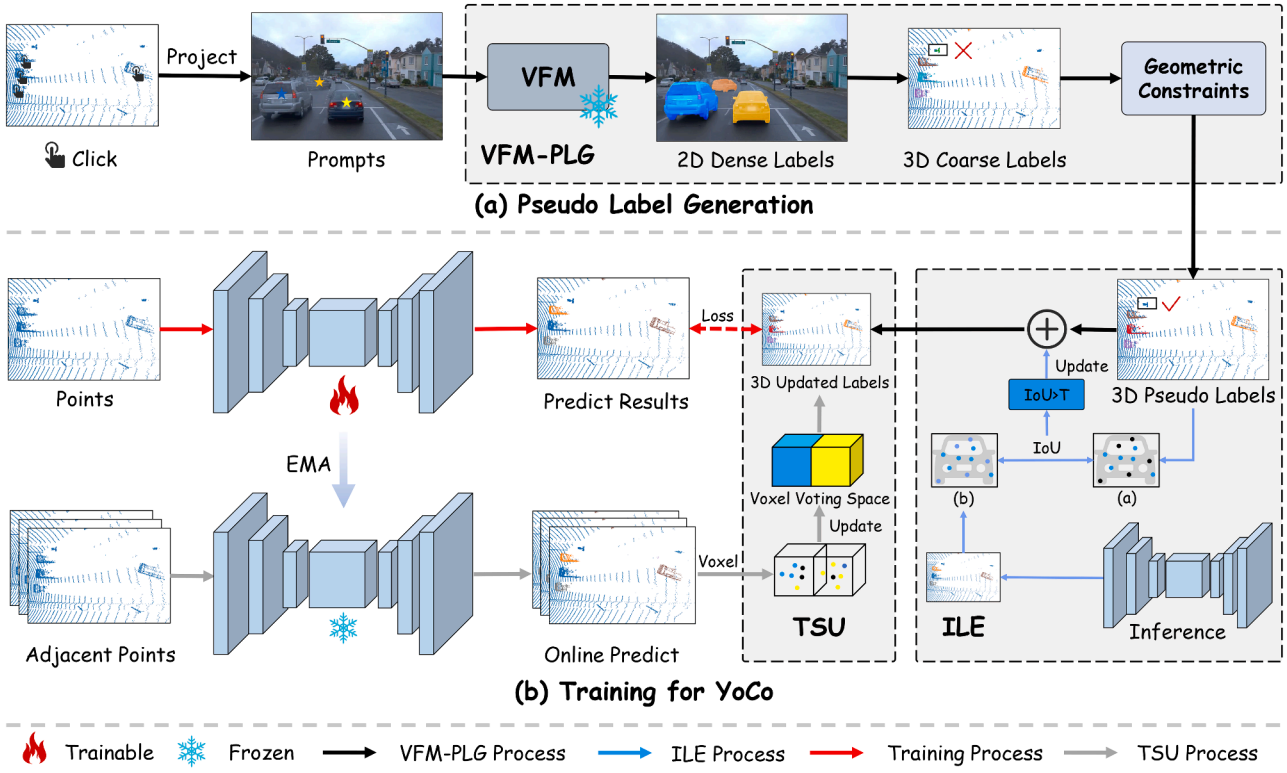


Fig. 2. Overview of the YoCo framework. The YoCo framework consists of two main components: (a) pseudo-label generation and (b) network training. In the pseudo-label generation step, the VFM-PLG module generates high-quality pseudo labels by leveraging VFMs combined with geometric constraints. For network training, YoCo adopts the classic Mean Teacher (Tarvainen & Valpola, 2017) structure. Initially, the VFM-PLG-generated pseudo labels are refined by the TSU module, which performs online updates by incorporating temporal consistency across adjacent frames. Once the TSU training is complete, the ILE module further enhances the pseudo labels offline by leveraging high-confidence and high-IoU predictions. These refined labels are then used for another round of TSU training, improving the label quality over time.

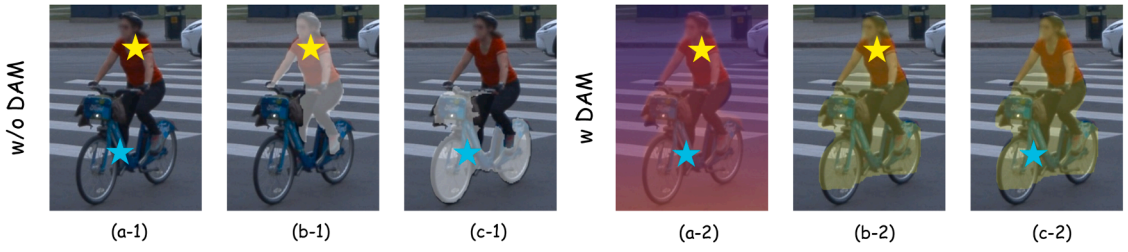


Fig. 3. VFM-PLG for the composite categories. Comparison of 2D mask results for the same instance with different prompts (colored stars). *w/o* shows results without DAM, while *w* shows results with DAM. Using the DAM model yields more consistent and accurate results across different prompts.

label m_i for the i-th object through the projection relationship. Then a clustering algorithm is applied to m_i, resulting in a set of clusters. The cluster containing the click annotation p_i is identified as the 3D pseudo label for that object ($Find$ operation in Eq. (4)).

Next, a geometric consistency check is applied to the identified cluster ($Filter$ operation in Eq. (4)). The label is retained if the cluster satisfies certain geometric conditions; otherwise, the pseudo label is discarded, *i.e.*, $m_i = \{0\}_{i=1}^{N}$. Specifically, for each category, we set a volume threshold to discard labels whose bounding box volumes exceed the specified threshold, thus maintaining consistent object scales in the pseudo labels. This process can be represented as:

$$\tilde{m}_i = Filter\big(Find\big(Cluster\big(m_i\big)\big)\big) \tag{4}$$

If the current click corresponds to multiple points, we iterate to select randomly one as the prompt. If the current result meets the geometric constraints, it is retained as the 3D pseudo label; otherwise, a new point is re-selected as the prompt to generate the pseudo label, as indicated by the blue dashed line in Fig. 4. This random selection helps explore different points in the 3D space and prevents bias in label generation.

3.3. Temporal and spatial-based label updating

To improve the quality of pseudo labels generated by the VFM-PLG module, we propose a temporal and spatial-based label updating module. This module leverages the generalization of neural networks by using high-reliability predictions from adjacent frames to update the pseudo labels of the current frame online. To transform the point clouds from adjacent frames to the current frame, a coordinate system transformation is required, which can be expressed by the following equation:

$$\mathbf{P}_t = \mathbf{T}_t^{-1} \times \mathbf{T}_{adj} \times \mathbf{P}_{adj} \tag{5}$$

where $\mathbf{T}_t$ is the ego-car pose in the current frame, $\mathbf{T}_{adj}$ is the ego-car pose in the adjacent frame, $\mathbf{P}_t$ and $\mathbf{P}_{adj}$ correspond to the coordinates of the point cloud in the current frame and the adjacent frame, respectively.

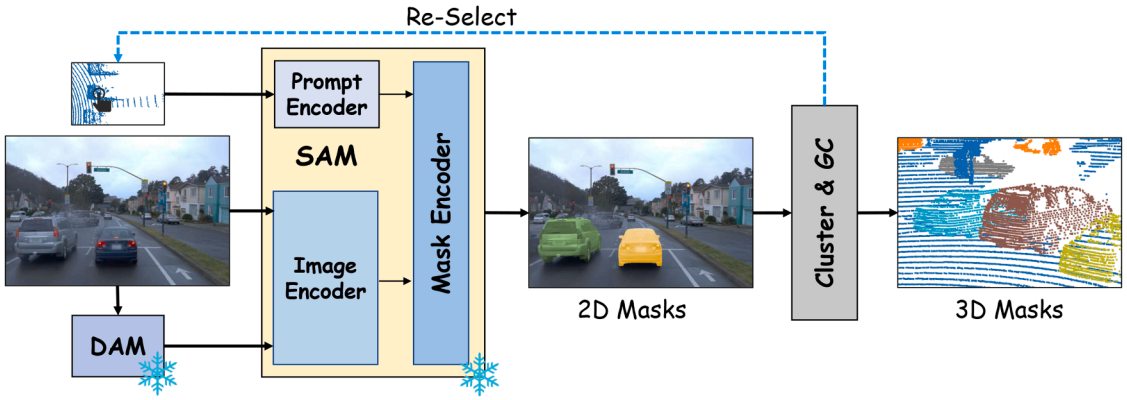


Fig. 4. Overview of VFM-PLG module. The blue dashed line indicates that if the generated 3D mask does not satisfy geometric constraints, another point is selected as the prompt. *GC* denotes that the point cloud is processed using geometric constraints.

Unlike MWSIS's (Jiang et al., 2024) PVC module, our method does not require establishing a voting space from the previous training epochs, which reduces memory requirements during training. Moreover, since it is difficult to establish a one-to-one correspondence between point clouds of adjacent frames, we employ a voxel voting mechanism. An online voxel voting space S is set up, where each voxel updates its corresponding label according to a predefined update strategy. The current frame requires voxelization of the point cloud, and the updated labels are obtained from the corresponding voxel space. The specific update strategy is as follows:

(1) Soft voting strategy. Consider a voxel that contains n points, in which each point p_i^v has an associated classification confidence scores $s_i^v \in \mathbb{R}^{num}$, where the *num* denotes the class number. We average the classification confidence scores for all points within the voxel, and then identify the class c_i^v with the highest score. If this score exceeds a set threshold T_s, the class is assigned as the voxel's label. The above process can be represented by Eq. (6). Unlike the method of directly selecting the class with the most points, this approach enhances robustness against prediction noise.

$$c_i^v = \begin{cases} \arg\max \bar{s} & \text{if } \max \bar{s} > T_s \\ -1 & \text{otherwise} \end{cases} \tag{6}$$

where $\bar{s} = 1/n \sum_{i=1}^{n} s_i^v$, -1 denotes the ignored label.

(2) Distance-based reliability update strategy. To improve the reliability of voting labels, we consider that a greater number of voxel points and higher point confidence result in more reliable voting. Given that LiDAR point clouds are dense near the sensor and sparse farther away, we dynamically adjust the voting threshold: voxels closer to the sensor require more votes and higher confidence for label assignment.

By applying the update strategy, a reliable voxel voting space S is constructed, which is then used to update the labels of the current frame. For further details, refer to Algorithm 1.

Discussion: From a conceptual perspective, the TSU module introduces temporal self-distillation into the weakly supervised training process. Adjacent frames in driving sequences inherently exhibit strong temporal-spatial consistency. By aligning and integrating high-confidence predictions from neighboring frames, TSU smooths out random fluctuations in pseudo labels, thereby mitigating label noise and enhancing supervision stability. Furthermore, the distance-adaptive mechanism ensures that reliable updates are propagated even in sparse regions, preventing overfitting to noisy labels. Functionally, TSU acts as a temporal regularizer, enforcing label continuity across frames and leading to smoother optimization.

Algorithm 1: TSU.

Input:
s_v is voxel size.
D is the depth threshold.
S is the voxel voting space.
Y is the current frame pseudo label.
T_v is the threshold for the number of votes.
T_s is the threshold for the confidence scores.
$(v_{x_e}, v_{y_e}, v_{z_e})$ is the ego voxelization coordinates.
$V_a = \{v_a : \{(x_a, y_a, z_a, s_a)\}_i^n\}$ is adjacent frame point voxelization dictionary, where v_a is the set of n points whose current voxelization coordinates are $(v_{x_a}, v_{y_a}, v_{z_a})$, and each set contains the coordinates (x_a, y_a, z_a) and confidence scores s_a for the current voxel v_a.
$V_t = \{v_t : \{(x_t, y_t, z_t, s_t)\}_i^{n'}\}$ is current frame point voxelization dictionary.
Output:
Updated pseudo label $\hat{Y}$.
for v_a *in* V_a **do**
  `// 1.Update Threshold.`
  $dist = s_v \sqrt{(v_{x_a} - v_{x_e})^2 + (v_{y_a} - v_{y_e})^2}$;
  $T'_{vote} = D/dist \times T_{vote}$;
  $T'_{score} = D/dist \times T_{score}$;
  `// 2.Build Voxel Voting Space.`
  $\bar{s} = \frac{1}{n}(\sum s_a)$;
  if $\max(\bar{s}) \geq T'_{score}$ *and* $n \geq T'_{vote}$, **then**
    $S\left[v_{x_a}, v_{y_a}, v_{z_a}\right] = \arg\max_{cls}(\bar{s})$;
  end
  else
    $S\left[v_{x_a}, v_{y_a}, v_{z_a}\right] = -1$;
  end
end
`// 3.Update Current Pseudo Label.`
$\hat{Y} = S\left[v_{x_t}, v_{y_t}, v_{z_t}\right]$;
$Mask = \hat{Y} == -1$;
$\hat{Y}[Mask] = Y[Mask]$.

3.4. IoU-guided label enhancement

To further leverage the robustness of neural networks and correct the erroneous pseudo labels generated by the VFM-PLG, we introduce an IoU-guided label enhancement module. This module performs an

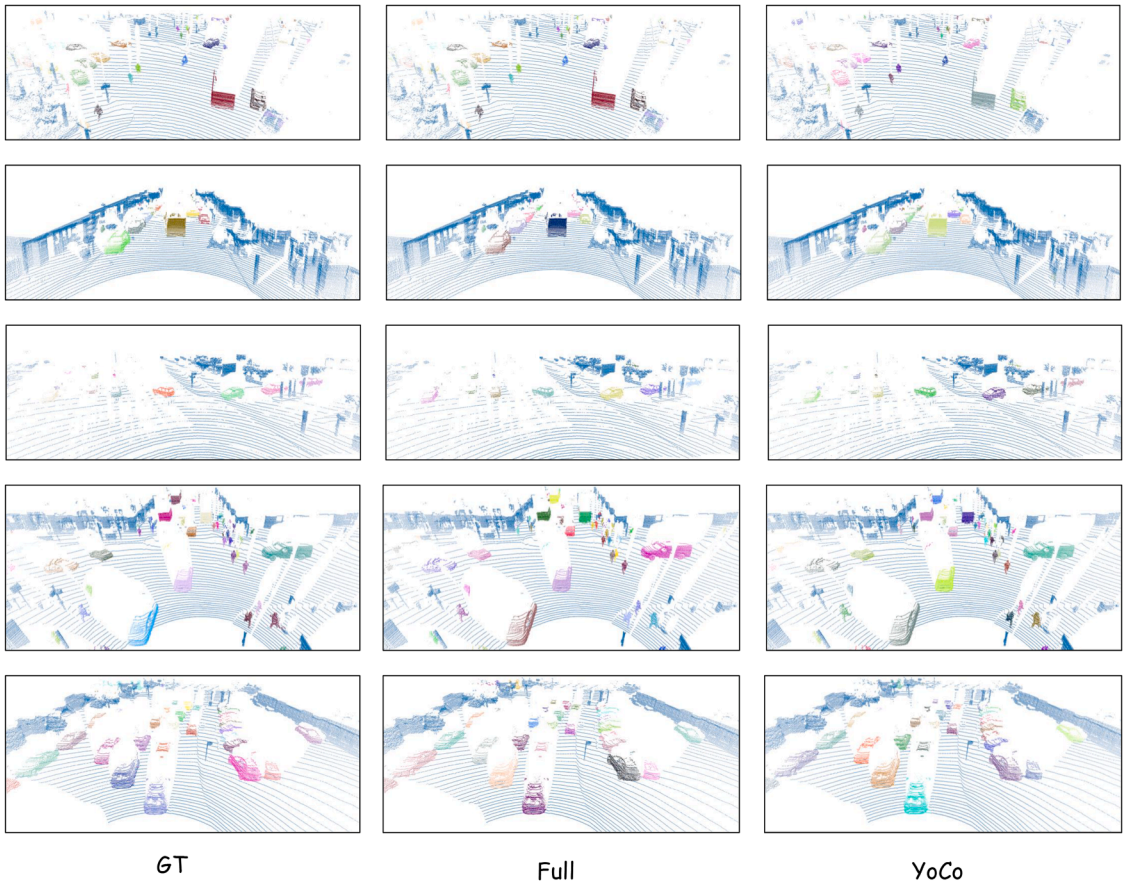


Fig. 5. Visualization for different scenes.

offline update of pseudo labels using high-confidence scores and high-IoU value predictions. Additionally, we adjust the confidence score threshold accordingly to accommodate the characteristic density variation in LiDAR point clouds (dense near, sparse far). The process can be represented as follows:

$$m_i' = \begin{cases} \arg\max\left(IoU_i\right) & \text{if } s_i \geq T_{s_2} \text{ and } IoU_i \geq T_{IoU} \\ \tilde{m}_i & \text{otherwise} \end{cases} \tag{7}$$

where $IoU_i = (\tilde{m}_i \cap \hat{m}_i)/(\tilde{m}_i \cup \hat{m}_i)$, m_i' and $\hat{m}_i$ represent the updated labels and the predicted labels, respectively. T_{s_2} and T_{IoU} correspond to the predefined confidence scores threshold and IoU threshold.

Discussion: Conceptually, the ILE module enforces spatial self-consistency by allowing pseudo labels to evolve through reliable, high-IoU, and high-confidence predictions. This dual filtering mechanism acts as a selective correction process-removing noise-induced supervision errors while preserving high-quality labels. As a result, gradient noise during optimization is suppressed, leading to more stable updates. In essence, ILE introduces a self-evolutionary refinement loop that progressively enhances label precision, enabling the model to approximate fully supervised behavior even under weak supervision.

3.5. Loss

We employ two prediction heads: one for semantic segmentation, supervised by L_{cls}, and another for per-point center voting, supervised by L_{vote}. The final instance predictions are obtained by clustering the predicted voting centers. The overall loss function is formulated as:

$$L = \alpha_1 L_{cls} + \alpha_2 L_{vote} \tag{8}$$

where L_{cls} denotes the cross entropy loss, while L_{vote} represents the L1 loss. α_1 and α_2 are hyperparameters to balance loss terms.

To learn instance-level geometry, L_{vote} encourages each point to predict an offset vector toward the geometric center of its corresponding 3D instance. Given the ground-truth instance label, we first compute the centroid of each instance by averaging the coordinates of all points belonging to it. For every point, the network predicts an offset vector from its own coordinate to this centroid. The voting loss is then defined as the L1 distance between the predicted and ground-truth offsets. This supervision drives all points within the same instance to converge toward a consistent center, enabling the network to form compact and discriminative instance embeddings that facilitate more accurate clustering in the subsequent segmentation stage.

4. Experiments

4.1. Dataset

We conduct experiments on two LiDAR datasets specifically designed for autonomous driving: Waymo Open Dataset (WOD) (Sun et al., 2020) and nuScenes dataset (Caesar et al., 2020).

Following the SOTA MWSIS (Jiang et al., 2024), we conduct our experiments on version 1.4.0 of the WOD, which includes both well-synchronized and aligned LiDAR points and images. The WOD consists of 1150 sequences (over 200K frames), with 798 sequences for training, 202 sequences for validation, and 150 sequences for testing. For the 3D segmentation task, the dataset contains 23,691 and 5976 frames for training and validation, respectively. We specifically focus on the vehicle, pedestrian, and cyclist categories for evaluation.

As for the nuScenes dataset, a multi-modal dataset encompassing 1000 distinct scenarios, which are recorded with a 32-beam LiDAR, six surround-view cameras, and five radars, annotated at a 2Hz frequency. The dataset contains a total of 1000 scenes, with 700 scenes used for

Table 1
Performance comparisons of 3D instance and semantic segmentation on Waymo validation dataset. **Bold** indicates optimal performance, and underline indicates sub-optimal performance. * represents the pseudo label generated by SAM using the corresponding click annotation as prompts. † denotes the pseudo label generated by our VFM-PLG module. Abbreviations: vehicle (Veh.), pedestrian (Ped.), cyclist (Cyc).

Supervision	Annotation	Model	3D Instance Segmentation (AP)				3D Semantic Segmentation (IoU)			
			mAP	Veh.	Ped.	Cyc.	mIoU	Veh.	Ped.	Cyc.
Full	3D Mask	Cylinder3D (Zhu et al., 2021)	51.40	75.31	38.12	40.76	78.903	96.476	83.666	56.567
	3D Mask	PTv3 (Wu et al., 2024)	60.08	75.73	53.63	51.32	83.679	96.686	85.500	68.852
	3D Mask	SparseUnet (Shi et al., 2020)	59.26	80.25	56.95	40.59	79.505	96.675	81.906	59.933
Weak	3D Box	SparseUnet (Shi et al., 2020)	49.32	69.00	45.96	33.01	72.545	89.471	73.581	54.582
	2D Box	SparseUnet (Shi et al., 2020)	35.48	44.54	36.84	25.08	63.831	74.102	72.113	45.278
	2D Box*	SparseUnet (Shi et al., 2020)	45.12	64.06	40.06	31.23	75.571	93.418	77.982	55.312
	2D Box	MWSIS (Jiang et al., 2024)	48.42	61.45	45.23	38.59	75.898	90.369	78.996	58.329
	Click*	SparseUnet (Shi et al., 2020)	25.36	43.19	16.46	16.42	51.919	70.161	42.121	43.475
	Click†	Cylinder3D (Zhu et al., 2021)	45.12	60.72	36.52	38.11	70.068	76.653	79.123	54.428
	Click†	PTv3 (Wu et al., 2024)	46.59	62.59	38.12	39.06	72.776	**82.109**	77.056	59.163
	Click†	SparseUnet (Shi et al., 2020)	47.37	64.10	41.50	36.51	72.189	79.850	78.619	58.097
	Click†	**YoCo (Ours)**	**55.35**	**67.69**	**55.25**	**43.12**	**74.770**	81.136	**81.716**	**61.459**

training, 150 for validation, and the remaining 150 held out for testing. We specifically focus on the barrier, bicycle, bus, car, construction vehicle, motorcycle, pedestrian, traffic cone, trailer, and truck categories for experiments.

4.2. Implementation details

4.2.1. Click setting

For the click annotation, we use the average coordinates of each instance from the BEV plane as a reference and then select the nearest point to simulate the manual click. Meanwhile, in Table 8, we also simulate the results with the manual annotation error.

4.2.2. Evaluation metric

We adopt the same evaluation metrics as Jiang et al. (2024). For 3D instance segmentation, we use average precision (AP) across different IoU thresholds to assess performance, while for 3D semantic segmentation, we use mean intersection-over-union (mIoU) as the evaluation metric.

4.2.3. Training setting

We choose several classic backbones, including Cylinder3D (Zhu et al., 2021), SparseUnet (Shi et al., 2020), and Point Transformer V3 (PTv3) (Wu et al., 2024). Cylinder3D, PTv3, and SparseUnet are trained for 40, 50, and 24 epochs, respectively. All models are trained on 4 NVIDIA 3090 GPUs with a batch size of 8, using the AdamW (Loshchilov, 2017) optimizer.

4.2.4. Module configuration

VFM-PLG. The clustering algorithm used is Connected Component Labeling (CCL), with the connectivity distances set to 0.6 for Vehicle, 0.1 for Pedestrian, and 0.4 for Cyclist. The dimensions (Length, Width, Height, Volume) for each class are as follows: Vehicle: $(12\text{m}, 3\text{m}, 4\text{m}, 144\,\text{m}^3)$, Pedestrian: $(1.0\text{m}, 1.0\text{m}, 2.0\text{m}, 2.0\,\text{m}^3)$, and Cyclist: $(2.5\text{m}, 1.0\text{m}, 2.0\text{m}, 5.0\,\text{m}^3)$.

TSU. The voxel size is set to $s_v = (0.25\text{m}, 0.25\text{m}, 0.2\text{m})$, the depth threshold D is 50 m, the voting threshold T_v is 2, and the confidence threshold T_s is 0.6. Additionally, the minimum number of voting points is set to 1, and the minimum confidence required for a valid vote is 0.5.

ILE. The IoU threshold T_{IoU} is set to 0.7. The adaptive confidence threshold T_{s2} is defined as follows: when the point's distance to the ego-vehicle is greater than or equal to 30 m, T_{s2} is set to 0.5, and when the point's distance to the ego-vehicle is less than 30 m, T_{s2} is set to 0.6.

Table 2
Performance comparisons on nuScenes validation dataset. *fg* denotes that only foreground points participate in the evaluation, ignoring background points.

Sup.	Method	mAP	mIoU	mIoU (*fg*)
Full	SparseUnet (Shi et al., 2020)	**49.56**	68.056	78.548
	Cylinder3D (Zhu et al., 2021)	45.74	66.198	76.335
	PTv3 (Wu et al., 2024)	46.10	**71.972**	**81.264**
Weak	Click	19.35	37.686	58.067
	YoCo (Ours)	**33.35**	**55.554**	**68.267**

4.3. Main results

4.3.1. Results on WOD

We compare the YoCo with other weakly supervised and fully supervised methods for 3D instance segmentation. Considering both computational time and memory efficiency, we select SparseUnet (Shi et al., 2020) as the baseline for our experiments. Additionally, Table 15 presents the results of YoCo with different networks, further demonstrating the generality of our method. In Table 1, our YoCo achieves the best performance among weakly supervised methods. It outperforms the Click*-based approach with a 29.99% improvement in mAP and surpasses the SOTA method MWSIS by 6.93% mAP, while utilizing more cost-effective sparse click annotations. Additionally, compared with methods using 3D bounding boxes, which have higher annotation costs, our approach achieves a 6.03% mAP improvement. Moreover, our method outperforms fully supervised Cylinder3D by 3.95% mAP.

We also provide metrics for 3D semantic segmentation. Compared to the Click*-based method, our approach achieves a 22.851% improvement in mIoU. When compared to methods based on 3D bounding boxes, it achieves a 2.225% mIoU increase. Additionally, our method reaches 94.76% of the fully supervised performance.

In Fig. 5, we present visualized segmentation results for several scenes. Our method achieves visual quality comparable to fully supervised results and ground truth (GT).

4.3.2. Results on nuScenes

We compare the YoCo with other weakly supervised and fully supervised methods for 3D instance segmentation. As shown in Table 2, our method YoCo achieves a 14% improvement in mAP and a 17.868% improvement in mIoU compared to the Click-based method, demonstrating its effectiveness.

Table 3
All modules ablation.

Method			AP				IoU			
VFM-PLG	TSU	ILE	Veh.	Ped.	Cyc.	mAP	Veh.	Ped.	Cyc.	mIoU
–	–	–	43.19	16.46	16.42	25.36	70.161	42.121	43.475	51.919
✓	–	–	64.10	41.50	36.51	47.37	79.850	78.619	58.097	72.189
✓	✓	–	66.58	50.16	39.81	52.18	**81.733**	80.705	60.907	74.449
✓	✓	✓	**67.69**	**55.25**	**43.12**	**55.35**	81.136	**81.716**	**61.459**	**74.770**

Table 4
VFM-PLG ablation.

Annotation	Method	AP	
		mAP	Cyc.
2D Box	SAM	45.12	31.23
Click	SAM	25.36	16.42
	+ Cluster	40.19	24.45
	+ DAM	42.13	30.27
	+ GC	**47.37**	**36.51**

Table 5
Pseudo label quality comparison.

Method	SAM	mAP	mIoU
2D Box	✓	10.27	**63.113**
2D Click	✓	9.23	42.185
3D Click	✓	15.16	50.552
3D Click + VFM-PLG	✓	**23.95**	60.106

Table 6
Frames ablation.

Frames	0	1	2	3	5
mAP	47.37	47.52	**52.18**	51.07	48.17
mIoU	72.189	68.346	**74.449**	74.107	72.345

Table 7
Ablation study of voting strategy in the TSU module. *Hard* voting means that the predicted label of the majority class among the points within a voxel is selected as the updated label. In contrast, *Soft* voting assigns the updated label based on the class with the highest average confidence score across all points within the voxel. *Count* and *Score* represent the use of dynamic thresholds, *Count* dynamically adjusts the required number of votes, while *Score* dynamically adjusts the confidence score threshold for label updates.

Frames	Vote Mode	Dynamic		mAP	mIoU
		Count	Score		
0	-	-	-	47.37	72.189
2	Hard	-	-	48.54	73.138
		✓	-	48.68	73.215
	Soft	-	-	49.97	73.152
		✓	-	50.83	73.564
		✓	✓	**52.18**	**74.449**

4.4. Ablation study and analysis

4.4.1. Effect of all modules

Table 3 presents the ablation study of all proposed modules. When solely the VFM-PLG module is utilized, we observe a substantial performance improvement, with instance segmentation and semantic segmentation improving by 22.01% mAP and 20.270% mIoU, respectively. This demonstrates the effectiveness of generating pseudo labels by combining the VFM module with geometric constraints from point clouds. When the TSU module is introduced, the performance further improves by 4.81% mAP and 2.26% mIoU. This highlights the value of leveraging the neural network's generalization by using high-confidence predictions from adjacent frames to update the current frame's labels, improving the quality of the pseudo labels. Moreover, our proposed ILE module leverages the robustness of the network to perform offline refinement of the pseudo labels generated by the VFM-PLG. This approach yields significant improvements in label quality, with instance segmentation improving by 3.17% mAP and semantic segmentation by 0.321% mIoU. These results demonstrate the effectiveness of our framework in progressively refining pseudo labels and narrowing the performance gap between weakly supervised and fully supervised methods.

For a more detailed analysis of each component, please refer to the subsequent detailed ablation experiments.

4.4.2. Effect of the VFM-PLG

In Table 4, we conduct an ablation study to evaluate the impact of each component in the VFM-PLG module. The second row is the baseline performance, where the model is trained using labels generated from click annotations and the SAM. When a clustering algorithm is adopted to refine pseudo labels, the mAP improves substantially by 14.83%. Additionally, applying the DAM to handle composite categories, such as cyclists, can further improve performance by 5.82% mAP. When geometric constraints are incorporated, the mAP reaches 47.37%, surpassing methods that utilize 2D boxes as prompts. Combining these methods not only improves segmentation performance but also significantly reduces annotation costs, demonstrating the efficiency and practicality of our approach in weakly supervised 3D instance segmentation.

As shown in Table 5, we also conduct experiments generating pseudo-labels using click signals on images. By directly processing the validation set and comparing the results, we found that using 3D clicks significantly outperformed both 2D clicks and 2D bounding boxes. Notably, when combined with our VFM-PLG module, the performance reached 23.95% mAP. The primary reason for this superiority is that 3D clicks on the BEV plane correspond to real 3D object points. In our VFM-PLG module, we design a strict filtering mechanism: if the generated 3D pseudo-label contains the real 3D point, the label is retained; otherwise, it is directly discarded. This mechanism fundamentally ensures the association between the pseudo-label and the real 3D instance, significantly improving pseudo-label quality and thereby boosting overall performance.

In Fig. 6, we evaluate the performance of SparseUnet trained with various annotation strategies. Our proposed method exhibits superior visual quality, significantly outperforming click-based approaches by generating more accurate predictions. This improvement is largely driven by the integration of deep prior information into the pseudo label generation process, which enhances the quality of the generated labels.

4.4.3. Effect of the TSU

Tables 6 and 7 provide the ablation studies for the TSU module. In Table 6, we analyze the impact of using different numbers of frames for label updates. We observe a steady improvement in performance as the number of frames increases up to a certain point. Specifically, when updating labels using predictions from two adjacent frames, the method achieves its best performance, with 52.18% mAP and 74.449% mIoU. However, as the number of frames exceeds three, the increased motion

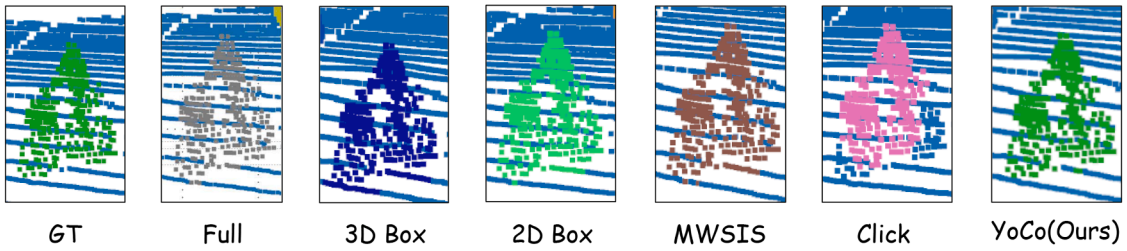


Fig. 6. Visualization for cyclist.

Table 8
Ablation study of the manual annotation error. The first column of the table presents the click error range during manual annotation. Single click annotations are randomly selected from a circular region centered at the instance's center in the BEV plane, with the *Error Range* defining the radius. A value of 0.0 indicates that the point closest to the instance center is selected.

Error Range (m)	mAP	mIoU
0.0	**55.35**	74.770
0.1	54.86	75.688
0.2	54.98	75.591
0.3	54.39	**75.936**
0.5	54.76	74.444

of dynamic objects causes greater discrepancies between frames, leading to erroneous votes and a consequent drop in overall performance.

Table 7 compares several voting strategies to assess their impact on segmentation results. When employing the hard voting method (row 3), compared to the baseline method (row 1), we observe only modest improvements, with gains of 1.17% mAP and 0.949% mIoU. This limited improvement suggests that the hard voting is not sufficiently robust to handle noise in the predictions. In contrast, when applying the soft voting method (row 4), the performance improvement is more pronounced, yielding 2.6% mAP and 0.963% mIoU gains.

Furthermore, when integrating our distance-based reliability update strategy (row 6), which adapts to the varying density of LiDAR point clouds based on their proximity to the sensor, we observe a significant increase in performance. Specifically, instance segmentation improves by 4.81% mAP, while semantic segmentation sees a gain of 2.260% mIoU. These results highlight the robustness of our proposed strategy in leveraging the inherent properties of LiDAR point clouds, particularly the denser distribution of points near the sensor and the sparser distribution at greater distances, to enhance the reliability of pseudo labels during training. This approach not only improves segmentation accuracy but also addresses challenges posed by noisy or uncertain predictions.

4.4.4. Computational efficiency analysis

The computational efficiency comparison shown in Table 9 demonstrates that the increase is minimal across all tested models, while there is a slight increase in both the training time and GPU memory usage when YoCo is applied. For example, the training time increases by approximately 1.2 hours (from 5.6 to 6.8 hours) for SparseUnet, 3.4 hours (from 16.7 to 20.1 hours) for Cylinder3D, and 5.0 hours (from 29.2 to 34.2 hours) for PTv3. Similarly, the memory usage increases by only 0.1-0.2 GB across all models. Notably, the inference time remains unchanged, indicating that YoCo does not introduce any additional latency during inference, maintaining a lightweight and efficient runtime during deployment.

This slight increase in training time and memory usage is expected, given the additional refinement modules and temporal updates introduced by YoCo. However, we believe that the small cost in training time and memory is justified by the significant reduction in annotation effort that YoCo offers, making it a highly efficient solution in terms of human labor and labeling costs.

4.4.5. Effect of the manual annotation error

To simulate potential errors that may arise during manual annotation, we conduct experiments with varying click annotation ranges, as detailed in Table 8. These results demonstrate that our method exhibits strong stability across different annotation ranges. Notably, even as the click range expands to 0.5 m, there is no significant drop in performance, with both mAP and mIoU remaining well within acceptable limits. This robustness indicates that our approach is resilient to variations in the click annotation radius, consistently maintaining high performance regardless of the distance from the instance center. Such stability is crucial in real-world applications, where manual annotations can introduce variability, yet reliable segmentation results are still required.

As shown in Table 8, the mIoU achieves the highest value when the click error range is set to 0.3 m. A similar trend can be observed in Table 10, where the AP for vehicle instances improves as the click range increases up to 2.0 m. These seemingly counterintuitive results share the same underlying cause. In the BEV space, the BEV center of an instance is not always the optimal click location. Points near the BEV center, especially for large objects such as vehicles, often correspond to the roof or bottom region, which, when projected onto the image plane, provides weak visual cues for the SAM model. Expanding the click range allows the annotation to fall on more informative surfaces, such as the sides or front of the object, enabling SAM to generate more accurate 2D masks and consequently higher-quality 3D pseudo labels.

Moreover, when suboptimal 2D masks are produced, our VFM-PLG module filters out unreliable masks through geometric constraints, while the TSU and ILE further refine them using temporal consistency and iterative label correction. These mechanisms collectively explain why performance can improve with moderate click perturbations (for example, 0.3 m for mIoU and up to 2.0 m for large vehicles) but eventually degrades when the error range becomes too large. The results indicate that the BEV center is not necessarily the best click location and confirm that YoCo is highly robust to annotation imprecision, consistently maintaining reliable segmentation quality even under imperfect supervision.

4.4.6. YoCo with false positives (FP) click

We consider that some small objects (such as *Construction Cone/Pole*) may be mislabeled as people in the annotation. To address this issue, we propose incorporating 2D semantic information to improve segmentation accuracy. Specifically, we intentionally introduce *Construction Cone/Pole* objects as *Pedestrian* noise in each scene to simulate real-world mislabeling scenarios. As demonstrated in Table 11, our method achieves 53.48% mAP and 74.936% mIoU when using GT boxes, which is comparable to the performance of YoCo. Furthermore, even with imprecise predictions from YOLO, our approach maintains robust

Table 9
Comparison of Computational Efficiency with and without YoCo.

Model	Training (h)		Memory (GB)		Inference (ms)	
	w/o YoCo	w YoCo	w/o YoCo	w YoCo	w/o YoCo	w YoCo
SparseUnet(Shi et al., 2020)	5.6	6.8	2.7	2.8	319	319
Cylinder3D (Zhu et al., 2021)	16.7	20.1	7.1	7.3	569	569
PTv3 (Wu et al., 2024)	29.2	34.2	20.4	20.6	798	798

Table 10
Greater click range error for the vehicle.

Range	0.0	0.1	0.5	1.0	1.5	2.0
AP	67.69	67.28	67.70	67.22	67.60	**68.06**
IoU	81.136	82.288	82.229	84.350	**85.400**	83.885

Table 11
YoCo with FP problem.

Box	GT	YOLO
mAP	**55.12**	53.48
mIoU	**75.120**	74.936

Table 12
Ablation study of drop click.

	Method	Proportion	mAP	mIoU
(a)	YoCo	16.31%	**55.35**	**74.770**
(b)	Discard Object	24.65% (10%)	54.29	73.742
(c)		37.19% (25%)	53.96	73.248
(d)		58.10% (50%)	46.68	70.008

Table 13
Ablation study of across sequences.

	Interval	mAP	mIoU
(a)	1	**55.35**	**74.770**
(b)	2	53.90	72.999
(c)	3	53.13	72.304

Table 14
Performance comparison of supervision strategies on the Waymo validation dataset. YOLO refers to pseudo labels derived from YOLO prediction results.

Supervision	Annotation	Model	mAP	mIoU
Full	3D Mask	SparseUnet	59.26	79.505
Weak	Click*	SparseUnet	40.19	67.510
	Click†	YoCo	55.35	74.770
Unsupervised	YOLO	YoCo	45.78	72.182

Table 15
Generality experiment of the YoCo.

Supervision	Annotation	Model	YoCo	mAP
Full	3D Mask	Cylinder3D (Zhu et al., 2021)	–	51.40
	3D Mask	SparseUnet (Shi et al., 2020)	–	59.26
	3D Mask	PTv3 (Wu et al., 2024)	–	60.08
Weak	Click†	Cylinder3D (Zhu et al., 2021)	–	45.12
	Click†	SparseUnet (Shi et al., 2020)	–	47.37
	Click†	PTv3 (Wu et al., 2024)	–	46.59
	Click†	Cylinder3D (Zhu et al., 2021)	✓	49.71
	Click†	SparseUnet (Shi et al., 2020)	✓	55.35
	Click†	PTv3 (Wu et al., 2024)	✓	53.83
Fine-Tune	0.8% 3D Mask	Cylinder3D (Zhu et al., 2021)	✓	53.13
	0.8% 3D Mask	SparseUnet (Shi et al., 2020)	✓	59.27
	5.0% 3D Mask	PTv3 (Wu et al., 2024)	✓	60.44

performance, validating the efficacy of leveraging 2D semantic information for verification.

4.4.7. Discard click analysis

To verify whether all foreground objects need to be clicked, we conduct a random discard experiment. It is noteworthy that our default method inherently discards objects with insufficient point cloud representations in the BEV perspective, resulting in a 16.31% object discard rate. We further evaluate the impact of additional discarding by randomly removing instances with probabilities of 10%, 25%, and 30%, as detailed in Table 12 (b)–(d). The experimental results indicate that as the discard rate increases, the network performance gradually declines. However, even with a discard proportion of 37.19%, our method still achieves 53.96% mAP and 73.248% mIoU, demonstrating the robustness of YoCo.

4.4.8. Scalability across sequences

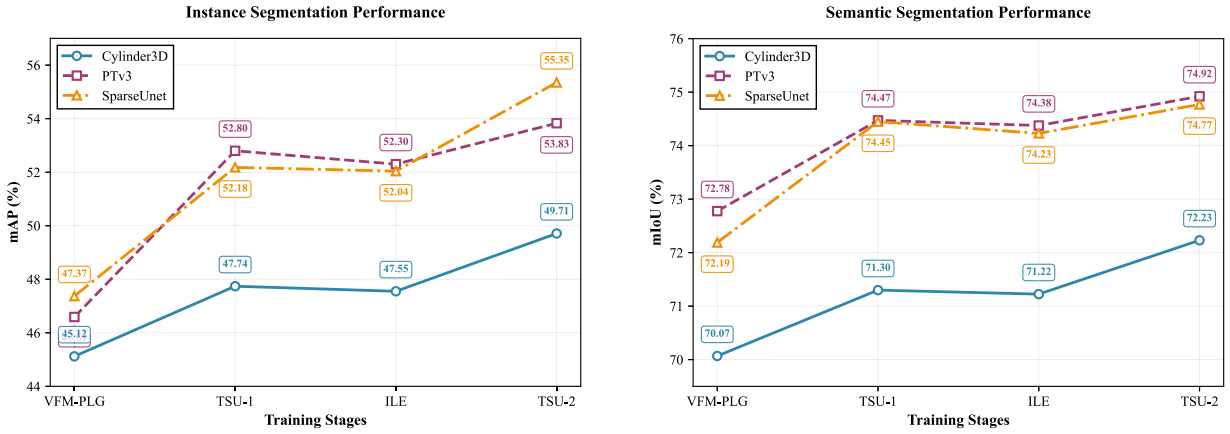
In Table 13, we assess the efficacy of YoCo under limited annotation scenes by subsampling point cloud sequences at intervals. Specifically, in Table 13 (b) and (c), we report performance metrics using 1/3 and 1/2 of the annotated sequences, respectively. Notably, even with only 1/3 of the annotation data, our method achieves 53.13% mAP and 72.304% mIoU, highlighting its scalability and robustness in resource-constrained settings.

4.4.9. Unsupervised application of YoCo

To extend the applicability of our approach, we investigate the use of the 2D detection model YOLO for unsupervised point cloud segmentation tasks. Specifically, we align YOLO's detection categories with the annotation categories of the Waymo dataset. Since YOLO does not directly support the *Cyclist* category, we propose a heuristic strategy: the *Cyclist* category is inferred by combining detections from the intersection of the *Person* and *Bicycle* categories. As shown in Table 14, our unsupervised approach achieves significant performance improvements, with a 5.59% increase in mAP and a 4.672% increase in mIoU compared to the click-based method. These results validate the effectiveness of leveraging 2D detection models for unsupervised point cloud segmentation tasks.

4.4.10. Fine-tuning with the YoCo

Following Jiang et al. (2024), we fine-tune the trained network within our YoCo framework across different proportions of fully supervised data. The 0.8% fully supervised data used for fine-tuning is randomly sampled. As shown in Fig. 1, the fine-tuning with a mere 0.8% of the fully supervised data achieves performance comparable to that obtained with full supervision. Moreover, when increasing the fully supervised data utilization to 5%, our approach surpasses the fully supervised performance of the SOTA PTv3. The above experiments demonstrate that our YoCo framework achieves performance comparable to fully supervised methods using only a minimal amount of annotations, significantly reducing annotation costs.



(a) Instance segmentation performance (b) Semantic segmentation performance

Fig. 7. Dynamic performance curves of three representative models (Cylinder3D, PTv3, and SparseUnet) under the proposed YoCo framework. VFM-PLG and ILE represent training using the pseudo-labels generated by VFM-PLG and ILE respectively, and TSU-1 and TSU-2 represent online update training based on the previous model.

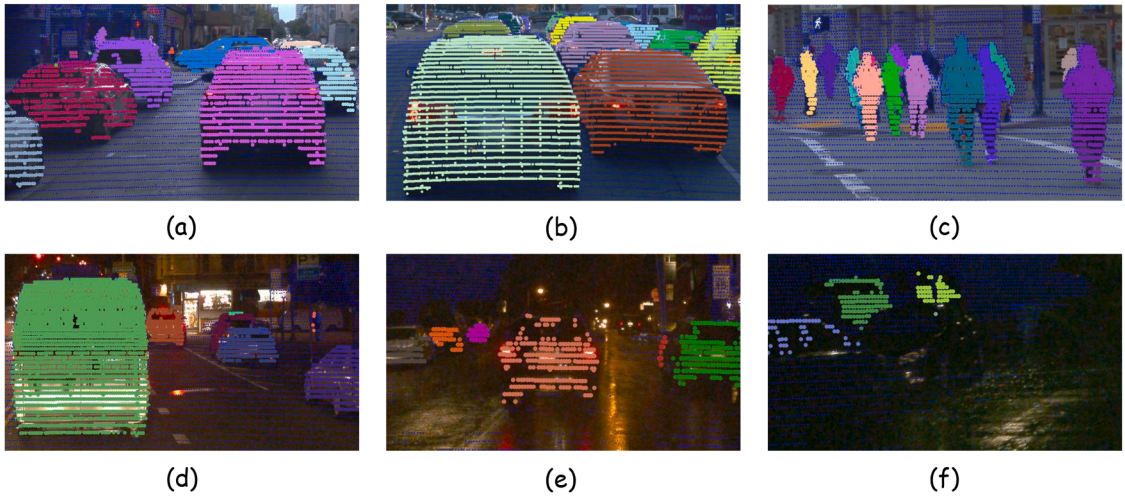


Fig. 8. Visiualization for challenging cases.

4.4.11. *The generality of the YoCo*

In Table 15, we present experiments on various networks, including Cylinder3D, SparseUnet, and PTv3, within the YoCo framework to validate the generality of our method. As shown, the YoCo exhibits strong performance across all three networks under weak supervision. Specifically, for the 3D instance segmentation task, our method improves mAP by 4.59% for Cylinder3D, 7.24% for SparseUnet, and 7.98% for PTv3. In addition, we fine-tune different networks trained under the YoCo using a small portion of fully supervised data. The results indicate that with only 0.8% of fully supervised data for fine-tuning, both Cylinder3D and SparseUnet surpass their fully supervised performance. When 5% of labeled data is used, the PTv3 network also exceeds its fully supervised performance. These results reveal that YoCo is not only effective for a single network architecture but also adaptable to multiple architectures, demonstrating its generality.

4.4.12. *Dynamic performance curves of the YoCo*

Fig. 7 visualizes the dynamic performance of Cylinder3D, PTv3, and SparseUnet within the proposed YoCo framework. Panel (a) shows that mAP increases steadily from VFM-PLG to TSU-1, then slightly drops after the ILE stage, and finally rises again at TSU-2 to achieve the best overall results. A similar trend is observed in Panel (b) for mIoU, where performance improves during the early stages, temporarily declines at ILE, and subsequently recovers at TSU-2. This fluctuation can be attributed to the offline refinement mechanism of the ILE module, which generates cleaner but less comprehensive pseudo labels by retaining only high-confidence and high-IoU predictions. Although this process momentarily affects the performance, it produces higher-quality supervision that benefits the subsequent TSU-2 stage, where temporal-spatial updates further calibrate the network and lead to consistent and stable improvements. These curves complement the tabular results by revealing how each stage contributes to convergence and demonstrates the progressive effectiveness of our learning strategy.

4.4.13. *YoCo for challenging cases*

To further provide a comprehensive evaluation, we include additional visualization results of challenging scenarios in Fig. 8, covering severe occlusion, dense traffic, and low-light/night environments. Subfigures (a-e) demonstrate that YoCo maintains high-quality segmentation performance under complex and cluttered scenes, showing its robustness to occlusion and dense objects. However, as shown in subfigure (f), the segmentation quality degrades notably in extremely dark scenes.

To analyze this failure case in detail, Fig. 9 visualizes the 2D masks generated by SAM for the same scene. Under such low illumination, SAM fails to accurately segment the target region, producing an incorrect 2D mask. Since our VFM-PLG module relies on SAM outputs and selects

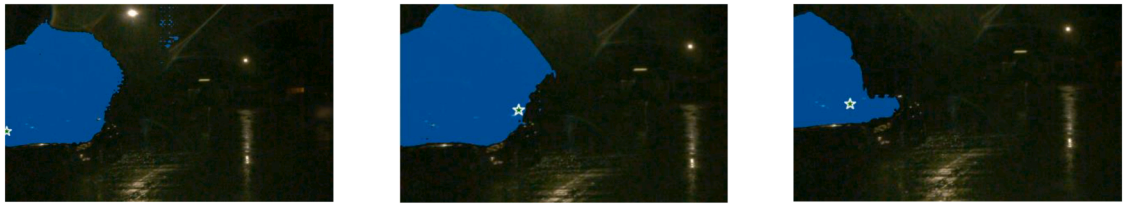


Fig. 9. Visualization for failure cases. The stars represent the positions where the click signal is projected onto the 2D image, and the blue areas represent the 2D masks of SAM segmentation.

the largest geometrically consistent cluster as the 3D pseudo label, this segmentation error propagates into the 3D space, leading to inaccurate instance masks.

This issue mainly stems from the inherent limitations of current 2D image segmentation models in handling extremely low-light conditions. Future research could explore employing more powerful vision foundation models with improved robustness and incorporating 3D geometric cues to mitigate the influence of imperfect 2D masks, thereby enhancing pseudo-label reliability in adverse environments.

5. Discussion and Limitations

The YoCo demonstrates strong potential for label-efficient 3D instance segmentation in autonomous driving scenarios. By leveraging vision foundation models and incorporating geometric, temporal, and instance-level cues, it achieves high-quality pseudo labels from only sparse click-level annotations. The method significantly reduces annotation costs while achieving competitive or even superior performance compared to fully supervised baselines. Moreover, its modular design allows generalization across different backbone architectures and datasets.

Despite these advantages, the YoCo's performance still depends on the quality of the underlying vision foundation models, particularly in 2D segmentation and depth prediction. Additionally, the method assumes accurate alignment between point clouds and RGB images; calibration or synchronization errors in the dataset may affect label quality. These limitations are peripheral to the core contributions and may be further addressed by future advances in vision models and sensor fusion techniques.

6. Conclusion

In this paper, we introduce YoCo, a novel framework for LiDAR point cloud instance segmentation using only click-level annotations. YoCo aims to minimize the performance gap between click-level supervision and full supervision. We achieve this through two key components: the VFM-PLG module, which generates high-quality pseudo labels using the zero-shot capability of the VFM model combined with geometric constraints from the point cloud, and the TSU and ILE modules, which refine labels by utilizing the robustness and generalization ability of neural networks. Our extensive experiments demonstrate that YoCo not only outperforms previous weakly supervised methods but also surpasses fully supervised methods based on the Cylinder3D, significantly reducing labeling costs while maintaining high segmentation performance. Additionally, our framework exhibits strong generality, making it applicable to various networks. These results highlight the efficiency and robustness of our approach, offering a practical solution for reducing annotation overhead in large-scale point cloud segmentation tasks.

CRediT authorship contribution statement

Guangfeng Jiang: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Visualization; **Jun Liu:** Conceptualization, Methodology, Writing – review & editing, Resources, Supervision, Project administration, Funding acquisition; **Yongxuan Lv:** Writing – review & editing, Software, Investigation; **Yuzhi Wu:** Writing – review & editing, Software, Investigation; **Xianfei Li:** Writing – review & editing, Visualization, Software, Investigation; **Wenlong Liao:** Writing – review & editing, Validation, Visualization; **Tao He:** Visualization, Validation, Data curation; **Pai Peng:** Writing – review & editing, Resources, Methodology, Funding acquisition.

Data availability

The authors do not have permission to share data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under contract 62471450, the Natural Science Foundation of Anhui Province under grant 2208085J17, and the Next-Gen Universal AI Robotic Brain (Robo-GPT): R&D and Applications .

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.eswa.2025.130824

References

Ando, A., Gidaris, S., Bursuc, A., Puy, G., Boulch, A., & Marlet, R. (2023). Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5240–5250).

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., & Gall, J. (2019). Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9297–9307).

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11621–11631).

Chen, Y., Xu, Z., Zhang, R., Jiang, X., Gao, X. et al. (2024). Foundation model assisted weakly supervised liDAR semantic segmentation. arXiv preprint arXiv:2404.12861.

Cheng, H., Zhu, J., Lu, J., & Han, X. (2024). Edgcnet: Joint dynamic hyperbolic graph convolution and dual squeeze-and-attention for 3d point cloud segmentation. *Expert Systems with Applications*, *237*, 121551.

Cheng, R., Razani, R., Taghavi, E., Li, E., & Liu, B. (2021). 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* (pp. 12547–12556).

Choy, C., Gwak, J., & Savarese, S. (2019). 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3075–3084).

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5828–5839).

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, *32*(11), 1231–1237.

Graham, B., Engelcke, M., & Van Der Maaten, L. (2018). 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9224–9232).

Guo, H., Zhu, H., Peng, S., Wang, Y., Shen, Y., Hu, R., & Zhou, X. (2024). Sam-guided graph cut for 3d instance segmentation. In *European conference on computer vision* (pp. 234–251). Springer.

He, X., Li, X., Xu, Q., Hu, Y., & Sun, Z. (2025). Radial awareness with adaptive hybrid CNN-transformer range-view representation for outdoor liDAR point cloud semantic segmentation. *Expert Systems with Applications*, *271*, 126572.

Hu, Q., Yang, B., Fang, G., Guo, Y., Leonardis, A., Trigoni, N., & Markham, A. (2022). Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds. In *European conference on computer vision* (pp. 600–619). Springer.

Jiang, G., Liu, J., Wu, Y., Liao, W., He, T., & Peng, P. (2024). Mwsis: Multimodal weakly supervised instance segmentation with 2d box annotations for autonomous driving. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 2507–2515).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al. (2023). Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4015–4026).

Kong, L., Liu, Y., Chen, R., Ma, Y., Zhu, X., Li, Y., Hou, Y., Qiao, Y., & Liu, Z. (2023). Rethinking range view representation for lidar segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 228–240).

Li, W., Liu, W., Zhu, J., Cui, M., Hua, R. Y. X., & Zhang, L. (2024). Box2mask: Box-supervised instance segmentation via level-set evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Liu, L., Kong, T., Zhu, M., Fan, J., & Fang, L. (2023). Clickseg: 3d instance segmentation with click-level weak annotations. arXiv preprint arXiv:2307.09732.

Liu, Z., Qi, X., & Fu, C.-W. (2021). 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* (pp. 1726–1736).

Loshchilov, I. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Ma, M., Han, X., Liang, S., Wang, Y., & Jiang, L. (2025). Connected vehicles ecological driving based on deep reinforce learning: Application of web 3.0 technologies in traffic optimization. *Future Generation Computer Systems*, *163*, 107544.

Meng, Q., Wang, W., Zhou, T., Shen, J., Jia, Y., & Van Gool, L. (2021). Towards a weakly supervised framework for 3d point cloud object detection and annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(8), 4454–4468.

Meng, Q., Wang, W., Zhou, T., Shen, J., Van Gool, L., & Dai, D. (2020). Weakly supervised 3d object detection from lidar point cloud. In *European conference on computer vision* (pp. 515–531). Springer.

Milioto, A., Vizzo, I., Behley, J., & Stachniss, C. (2019). Rangenet + +: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4213–4220). IEEE.

Ngo, T. D., Hua, B.-S., & Nguyen, K. (2023). Gapro: Box-supervised 3d point cloud instance segmentation using gaussian processes as pseudo labelers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 17794–17803).

Park, J., Kim, C., Kim, S., & Jo, K. (2023). Pcscnet: Fast 3d semantic segmentation of lidar point cloud for autonomous car using point convolution and sparse convolution network. *Expert Systems with Applications*, *212*, 118815.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652–660).

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017b). Pointnet + +: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, *30*.

Shi, S., Wang, Z., Shi, J., Wang, X., & Li, H. (2020). From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(8), 2647–2664.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B. et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2446–2454).

Tao, A., Duan, Y., Wei, Y., Lu, J., & Zhou, J. (2022). Seggroup: Seg-level supervision for 3d instance and semantic segmentation. *IEEE Transactions on Image Processing*, *31*, 4952–4965.

Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, *30*.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6411–6420).

Tsitsokas, D., Kouvelas, A., & Geroliminis, N. (2023). Two-layer adaptive signal control framework for large-scale dynamically-congested networks: Combining efficient max pressure with perimeter control. *Transportation Research Part C: Emerging Technologies*, *152*, 104128.

Unal, O., Dai, D., & Van Gool, L. (2022). Scribble-supervised lidar semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2697–2707).

Wang, S., Yu, J., Li, W., Shi, H., Yang, K., Chen, J., & Zhu, J. (2024). Label-efficient semantic scene completion with scribble annotations. arXiv preprint arXiv:2405.15170.

Wang, T., Ma, M., Liang, S., Yang, J., & Wang, Y. (2025). Robust lane change decision for autonomous vehicles in mixed traffic: A safety-aware multi-agent adversarial reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, *172*, 105005.

Wei, J., Lin, G., Yap, K.-H., Hung, T.-Y., & Xie, L. (2020). 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* (pp. 4384–4393).

Wu, W., Qi, Z., & Fuxin, L. (2019). Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9621–9630).

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., & Zhao, H. (2024). Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4840–4851).

Wu, X., Lao, Y., Jiang, L., Liu, X., & Zhao, H. (2022). Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, *35*, 33330–33342.

Xia, Q., Lin, H., Ye, W., Wu, H., Luo, Y., Zhao, S., Li, X., & Wen, C. (2024). Oc3d: Weakly supervised outdoor 3d object detection with only coarse click annotation. arXiv preprint arXiv:2408.08092.

Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337.

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10371–10381).

Yu, Q., Du, H., Liu, C., & Yu, X. (2024). When 3d bounding-box meets SAM: Point cloud instance segmentation with weak-and-noisy supervision. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 3719–3728).

Zeng, Z., Xu, Y., Xie, Z., Tang, W., Wan, J., & Wu, W. (2024). Large-scale point cloud semantic segmentation via local perception and global descriptor vector. *Expert Systems with Applications*, *246*, 123269.

Zhang, D., Liang, D., Zou, Z., Li, J., Ye, X., Liu, Z., Tan, X., & Bai, X. (2023). A simple vision transformer for weakly semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8373–8383).

Zhang, Y., Hu, Q., Xu, G., Ma, Y., Wan, J., & Guo, Y. (2022). Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18953–18962).

Zhao, H., Jiang, L., Jia, J., Torr, P. H. S., & Koltun, V. (2021). Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16259–16268).

Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., & Lin, D. (2021). Cylindrical and asymmetrical 3d convolution networks for liDAR segmentation. In *IEEE conference on computer vision and pattern recognition, CVPR 2021, Virtual, June 19–25, 2021* (pp. 9939–9948). Computer Vision Foundation / IEEE.