

# LWSIS: LiDAR-Guided Weakly Supervised Instance Segmentation for Autonomous Driving

Xiang Li<sup>1\*</sup>, Junbo Yin<sup>1\*</sup>, Botian Shi<sup>2</sup>, Yikang Li<sup>2</sup>, Ruigang Yang<sup>3</sup>, Jianbing Shen<sup>4†</sup>

<sup>1</sup>School of Computer Science, Beijing Institute of Technology

<sup>2</sup>Shanghai AI Laboratory

<sup>3</sup>Inceptio

<sup>4</sup>SKL-IOTSC, CIS, University of Macau

{lixianggoing, yinjunbo.cn}@gmail.com

## Abstract

Image instance segmentation is a fundamental research topic in autonomous driving, which is crucial for scene understanding and road safety. Advanced learning-based approaches often rely on the costly 2D mask annotations for training. In this paper, we present a more artful framework, **LiDAR-guided Weakly Supervised Instance Segmentation (LWSIS)**, which leverages the off-the-shelf 3D data, *i.e.*, Point Cloud, together with the 3D boxes, as natural weak supervisions for training the 2D image instance segmentation models. Our *LWSIS* not only exploits the complementary information in multimodal data during training, but also significantly reduces the annotation cost of the dense 2D masks. In detail, *LWSIS* consists of two crucial modules, Point Label Assignment (PLA) and Graph-based Consistency Regularization (GCR). The former module aims to automatically assign the 3D point cloud as 2D point-wise labels, while the latter further refines the predictions by enforcing geometry and appearance consistency of the multimodal data. Moreover, we conduct a secondary instance segmentation annotation on the nuScenes, named *nuInsSeg*, to encourage further research on multimodal perception tasks. Extensive experiments on the *nuInsSeg*, as well as the large-scale Waymo, show that *LWSIS* can substantially improve existing weakly supervised segmentation models by only involving 3D data during training. Additionally, *LWSIS* can also be incorporated into 3D object detectors like PointPainting to boost the 3D detection performance for free. The code and dataset are available at <https://github.com/Serenos/LWSIS>.

## Introduction

Instance Segmentation (He et al. 2017; Chen et al. 2019; Tian, Shen, and Chen 2020; Kirillov et al. 2020; Chen et al. 2020; Zhang et al. 2021) aim to recognize distinct instances of objects in an image by predicting pixel-level category and instance identity, which has benefited a wide range of applications such as robotics and autonomous driving. Popular image segmentation models (He et al. 2017; Tian, Shen, and Chen 2020; Chen et al. 2017) are often trained by pixel-level

\*These authors contributed equally.

†Corresponding author. Email: jianbingshen@um.edu.mo  
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

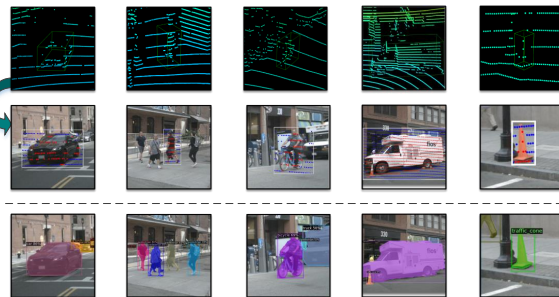


Figure 1: The basic idea of our *LWSIS* on *nuInsSeg* dataset. The first row shows the raw point clouds, the second row illustrates the assigned point label via the PLA module, and the third row demonstrates the segmentation results. Our *LWSIS* is trained without any mask annotations.

mask label. However, obtaining such fine-grained annotations can be very expensive and time-consuming, especially for self-driving vehicles, which typically require millions of training samples. In contrast, weakly supervised instance segmentation (Tian et al. 2021; Lan et al. 2021; Cheng, Parkhi, and Kirillov 2022; Lee et al. 2021; Wang et al. 2021), which tends to leverage cheaper and readily available annotations, has attracted increasing attention.

Several efforts have been made on the weakly supervised instance segmentation by utilizing box-level (Song et al. 2019; Tian et al. 2021) and point-level (Bearman et al. 2016; Cheng, Parkhi, and Kirillov 2022) annotations. However, all these works focus on the *single-modal* weak supervision. In fact, advanced self-driving vehicles are often equipped with both LiDAR and camera sensors to accurately capture the 3D and 2D scenes. Therefore, a smart way is to inherit the fruit of off-the-shelf LiDAR data and box annotations that are available on most autonomous driving datasets (Geiger, Lenz, and Urtasun 2012; Caesar et al. 2020; Sun et al. 2020), as well as to explore the *multimodal* weak supervision. This can largely save the annotation cost, eliminating the requirement of additional 2D mask-level annotations. Besides, by mining the geometrical information in 3D point cloud, the image segmentation results can be further improved. Our ba-

sic idea is illustrated in Fig. 1.

The main challenge in learning *multimodal* model is that the point clouds are relatively sparse and noisy, and the inaccurate calibration between LiDAR and camera sensors will corrupt the model performance. To address this, we propose a novel method that artfully mines the clues in LiDAR point cloud to guide the learning of Weakly Supervised Instance Segmentation (*LWSIS*) in images. Our *LWSIS* relies on two key ingredients: *Point-wise Label Assignment* (PLA) module and *Graph-based Consistency Regularization* (GCR) module, where the former module is used to assign LiDAR point cloud as point-wise pseudo annotations for the images and the latter module aims to further penalize incorrect segmentation predictions. More specifically, PLA contains four necessary steps to convert the LiDAR point cloud to point-wise pseudo labels, which are point cloud projection, depth-guided point refinement, label assignment and label propagation. To mitigate the confirmation bias of PLA, our GCR further enforces consistency regularization on the undirected graph built up on point cloud similarity. Our core idea is that 3D points with similar geometric and appearance features should have the same labels. Consequently, the instance segmentation model can be jointly optimized by the point-wise pseudo-labels generated by PLA and the graph-based regularization term given by GCR.

Moreover, our *LWSIS* is also a plug-and-play module that can be readily incorporated into existing weakly supervised instance segmentation models (Tian et al. 2021; Cheng, Parkhi, and Kirillov 2022) as an *auxiliary training task* to improve the model’s capability, requiring no extra network parameters and computation during inference. Since few self-driving datasets have provided accurate pixel-level instance segmentation labels that are synchronized with the 3D annotation due to the heavy annotation burden, we further contribute a secondary annotation for image instance segmentation based on nuScenes (Caesar et al. 2020), and name it as *nuInsSeg*. We adopt an efficient semi-automatic labeling method with human refinement to keep the segmentation at a high quality. *nuInsSeg* extends the nuScenes dataset with a large amount of 2D segmentation labels for 947K object instances. This also ensures an accurate and fair evaluation of our method with existing 2D instance segmentation models.

To summarize, we present a novel learning paradigm, *LWSIS*, that inherits the fruits of off-the-shelf 3D point cloud to guide the training of 2D instance segmentation models. This removes the dependency on mask-level image annotations. To our best knowledge, this is the first work that explores the *multimodal* weakly supervised instance segmentation. To realize this, the Point Label Assignment (PLA) module and the Graph-based Consistency Regularization (GCR) module are presented. Furthermore, We advocate a new dataset *nuInsSeg* based on nuScenes to extend existing 3D LiDAR annotations with 2D image segmentation annotations. The broad effectiveness of our *LWSIS* is demonstrated on the *nuInsSeg* and Waymo datasets. It shows our model is superior to other weakly supervised approaches and even surpasses the fully supervised models. The proposed *LWSIS* can also be readily applied to current 3D object detectors like

PointPainting (Vora et al. 2020) to further improve the 3D detection performance *for free*. We hope our *LWSIS* and *nuInsSeg* could help researchers to conduct better studies on multimodal perception in autonomous driving scenarios.

## Related Works

Weakly supervised instance segmentation aims to extract objects with simple and cheap annotations such as image-level tags (Ahn, Cho, and Kwak 2019; Cholakkal et al. 2019; Ge et al. 2019), points (Cheng, Parkhi, and Kirillov 2022; Lee, Kim, and Sull 2021), scribble (Tang et al. 2018) and bounding box (Tian et al. 2021; Arun, Jawahar, and Kumar 2020; Khoreva et al. 2017; Hsu et al. 2019; Lee et al. 2021) instead of expensive pixel-level annotations.

For methods using point annotations, PointSup (Cheng, Parkhi, and Kirillov 2022) proposes to use bounding box and random sampled points as segmentation annotations. It achieves performance close to 95% of fully supervised methods on large-scale COCO dataset. For methods supervised by bounding box annotations, BBTP (Hsu et al. 2019) proposed the first end-to-end trainable method with box supervision. They propose multiple instance learning (MIL) formulation to leverage tightness property of bounding box which assumes that a crossing line within a box will cover at least one pixel of the object. BBAM (Lee et al. 2021) used the attribution map from trained object detector which highlights the object regions. BoxInst (Tian et al. 2021) is a state-of-the-art box-supervised instance segmentation method which supervises the mask branch by a projection loss and a pair-wise similarity loss. Further, some methods introduce auxiliary tasks (Wang et al. 2021; Lan et al. 2021) to improve performance. BoxCaseg (Xu et al. 2021) takes both salient images and box annotations as supervision to perceive accurate boundary information. DiscoBox (Lan et al. 2021) is a multi-task method to solve instance segmentation and semantic correspondence simultaneously through weakly supervised joint training.

Basically, all above approaches are *single-modal* weakly supervised models. In this work, we make the first effort for *multimodal* weakly supervised instance segmentation model. It can bring further improvement to many existing weakly supervised models, and even surpass some fully supervised models like CondInst.

## The Proposed LWSIS Method

There are several advantages of exploring the *multimodal* weakly supervised segmentation model. First, the LiDAR point cloud has perceived the geometric shapes of interested objects and the projection of these points can serve as a natural supervision signal for training the image segmentation models. This also removes the need for additional 2D mask annotations. Second, the 2D image segmentation models could exploit the 3D geometrical features provided by the point cloud, to further improve the segmentation performance. Third, the obtained weakly supervised segmentation model can in turn promote 3D perception tasks like LiDAR-based object detection (Yin, Zhou, and Krahenbuhl 2021; Shi et al. 2020; Yin et al. 2021, 2022a,b; Meng et al.

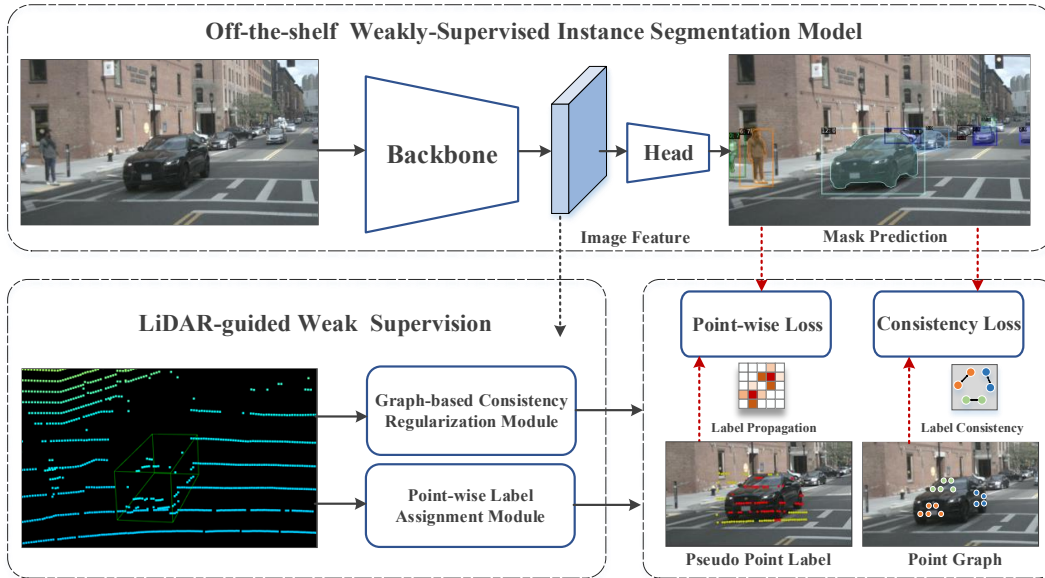


Figure 2: The overall architecture of our approach. The top image branch is an off-the-shelf weakly supervised instance segmentation model. The bottom branch is *LWSIS* which converts LiDAR point cloud into supervision for the image segmentation.

2020; Wang et al. 2023), through *multimodal* fusion. As for the 3D annotations, we just consider the ones that are already available in most datasets such as KITTI, nuScenes and Waymo, rather than annotating 3D data specifically for this 2D segmentation task. From a 3D perspective, the resulting 2D segmentation models can be seen as a free gift. All these merits motivate us to devise *LWSIS*.

The overview of our *LWSIS* is illustrated in Fig. 2, which consists of an image instance segmentation branch (top) and a point cloud weak supervision branch (bottom). Since our method can be readily integrated into off-the-shelf weakly supervised models, we choose BoxInst (Tian et al. 2021) and PointSup (Cheng, Parkhi, and Kirillov 2022) as examples in the top branch, which can produce initial instance segmentation predictions. Then, in the bottom branch, we convert LiDAR point cloud into weak supervision for optimizing the image instance segmentation predictions. To this end, we design the Point-wise Label Assignment Module (PLA) module and Graph-based Consistency Regularization (GCR) module. The PLA module takes LiDAR point cloud and 3D bounding boxes as the input and outputs the point pseudo labels for images. The GCR module exploits the similarity between neighboring points by a graph and regularizes the mask predictions with a consistency loss. Next, we detail the design recipes of these modules.

### Point-Wise Label Assignment Module

In order to take full advantage of the LiDAR point cloud, we design the PLA module to automatically assign each point a pseudo annotation to train the image segmentation model. To achieve this, the first step is to project the point cloud to the image plane. However, we find the parallax of the LiDAR sensor and the camera sensor will cause misalignment during the projection. Thus, a depth-guided refinement mod-

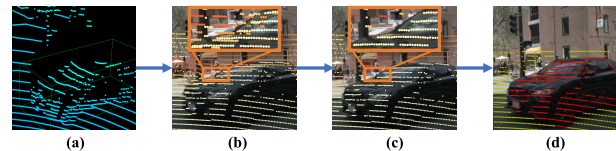


Figure 3: The overview of the PLA module. It takes as input (a) the point cloud and 3D box, and then projects them to (b) the camera coordinates, where the point color in (b) and (c) indicates the depth. The orange box in (b) shows the ‘noisy’ points caused by the inconsistent heights of LiDAR and camera, *e.g.*, the occluded points appear on the car. In (c), a depth-guided refinement method is designed to filter out these ‘noisy’ points. Finally, we assign (d) the rectified points as foreground (red) and background (yellow) labels.

ule is introduced to filter these noisy points. Then, we use a heuristic rule to assign the each refined point a binary label that represents the foreground or background. Finally, we further propagate these point-wise labels to neighbor pixels with similar features to provide dense supervision. The overview of the PLA is shown in Fig. 3.

**Point Cloud Projection.** The point cloud with  $N$  points in 3D space can be represented in homogeneous coordinate system as  $P_{3d} \in \mathbb{R}^{N \times 4}$ . The transformation matrix  $T_{(c \leftarrow 1)} \in \mathbb{R}^{4 \times 4}$  is used to project the point cloud from the LiDAR system to the camera system. Then we introduce the camera matrix  $M \in \mathbb{R}^{3 \times 4}$  to conduct the transformation from the camera to the image plane. Finally, the transformation from point cloud to image can be formulated as:

$$P_{2d}^T = MT_{(c \leftarrow 1)}P_{3d}^T \quad (1)$$

where  $P_{2d}^T \in \mathbb{R}^{N \times 3}$  is the projected point cloud in homogeneous coordinates.

**Depth-Guided Point Refinement.** In practical application of advanced self-driving vehicles, the LiDAR sensor is often installed at a higher position than the camera, which will lead to the parallax of the two sensors. As shown in Fig. 3(b), some projected points (*e.g.*, the orange point) are actually from the objects behind the car but are still visible on the car in the image space. This will cause noisy supervision signals in the label assignment process. To tackle this, we design the depth-guided point refinement method to remove these noisy points according to the assumption that the depth variation of an object should be smooth without a gradient cliff, *i.e.*, a point with larger depth will be viewed as a noisy point if its neighbor points all have smaller depth.

To be specific, given the projected 2D points  $P_{2d}$  on the image space, we first record the depth value of at each projected pixel to get a sparse depth map  $D \in \mathbb{R}^{H \times W}$ , where  $H$  and  $W$  is the size of the image. The pixel positions without projected points will be set to 0. Then, a 2D sliding window with a certain step size is utilized to process the depth map and remove the noisy points in each window. Specifically, the points inside each window are divided into two sets  $\mathcal{P}_{near}$ ,  $\mathcal{P}_{far}$ , according to the relative depth:

$$\mathcal{P}_{near} = \left\{ p(x, y) \mid \frac{d(x, y) - d_{min}}{d_{min}} < \tau_{depth} \right. \\ \left. d(x, y) \neq 0, \forall (x, y) \in \mathcal{W} \in \mathbb{R}^2 \right\} \quad (2)$$

where  $p(x, y)$  indicates the projected point at the pixel position  $(x, y)$  that falls within the local window  $\mathcal{W}$ , and  $\tau_{depth}$  is the pre-defined depth threshold that will remove points with relatively large depth.  $d(x, y)$  denotes the depth value at pixel position  $(x, y)$ , and  $d_{min}$ ,  $d_{max}$  are the minimum and maximum depth value within the window  $\mathcal{W}$ . Similarly, we can get  $\mathcal{P}_{far}$  if the relative depth exceeds  $\tau_{depth}$ . However, not all the distant points are necessary noise points. Thus we further calculate a minimum enclosing box formed by  $\mathcal{P}_{near}$ . The intuition is a valid point should have similar depth with its neighbor points. This can be denoted as:

$$\mathcal{P}_{noise} = \left\{ p(x, y) \mid x \in [x_{min}, x_{max}], \right. \\ \left. y \in [y_{min}, y_{max}], \forall p(x, y) \in \mathcal{P}_{far} \right\} \quad (3)$$

where  $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$  are the maximum and minimum values of points of the x and y axes in  $\mathcal{P}_{near}$ . Finally, the 2D points after refinement can be formulated as:

$$\mathcal{P}_{refine} = \mathcal{P}_{near} \cup \mathcal{P}_{far} \setminus \mathcal{P}_{noise} \quad (4)$$

**Label Assignment.** Here we introduce how to generate positive and negative point-wise labels. In particular, according to the positional relationship between the point cloud and the 3D detection bounding boxes, the  $\mathcal{P}_{refine}$  is further divided into two sets  $\mathcal{P}_{in}$  and  $\mathcal{P}_{out}$ . Since  $\mathcal{P}_{in}$  contains points inside a 3D box, a nature idea is to define  $\mathcal{P}_{in}$  as positive samples that represents foreground objects. Regarding  $\mathcal{P}_{out}$ , since its point number is extremely large, we only retain a subset of  $\mathcal{P}_{out}$  as negative samples, *e.g.*, we reserve the points around  $\mathcal{P}_{in}$  as hard examples. This also essentially ensures the balance of positive and negative samples.

More specifically, we first project the 8 vertexes of the 3D box to the image coordinates and then calculate the minimum enclosing rectangle  $b \in \mathbb{R}^{4 \times 2}$ , which can be regarded as a relaxed 2D bounding box. Then, we only keep the points inside  $b$  from  $\mathcal{P}_{out}$  and denote the resultant points set as  $\mathcal{P}'_{out}$ . The point-wise pseudo label is then given by:

$$l(p_i) = \begin{cases} 1 & , \text{ if } p_i \in \mathcal{P}_{in} \\ 0 & , \text{ if } p_i \in \mathcal{P}'_{out} \\ -1 & , \text{ otherwise} \end{cases} \quad \forall p_i \in \mathcal{P}_{refine} \quad (5)$$

where the binary label  $l(p_i)$  determines the point  $p_i$  to be a positive or negative sample. To further facilitate the batch-level learning during training, a fixed number of  $s$  points will be sampled from  $\mathcal{P}_{in}$  and  $\mathcal{P}'_{out}$  respectively with a certain positive and negative sample ratio. If  $s < |\mathcal{P}_{in} \cup \mathcal{P}'_{out}|$ , we will pad it by randomly sampling other point cloud through a Gaussian distribution. In this way, we obtain  $s$  points as the pseudo segmentation labels.

**Label Propagation.** Since the sampled points are inevitably sparse due to the nature of point cloud, we further propagate these pseudo labels to neighbor pixels with similar feature to provide dense supervision:

$$l(p_i) = \begin{cases} l(p_c) & , \text{ if } \exp(-f(p_i)f(p_c)) > \tau_d \\ -1 & , \text{ otherwise} \end{cases} \quad (6)$$

where  $l(p_i)$  is the assigned pseudo label for  $p_i \in \mathcal{N}_{p_c}$ , the neighbor pixels of candidate point  $p_c$ .  $f(p) \in \mathbb{R}^C$  means the image feature at position  $p$  that is extracted from the backbone.  $\tau_d$  is the similarity threshold, *i.e.*, we propagate the label of  $p_c$  to its neighbors only when the image feature similarity exceeds  $\tau_d$ . This leads to an enlarged set of pseudo point labels of number  $S$ .

**Point-Wise LiDAR Loss.** Instance segmentation models output mask-level predictions on regular grids, which can be represented as  $M \in \mathbb{R}^{h \times w}$ , where  $h, w$  is the prediction resolution. In our framework, we aim to optimize this mask prediction on  $S$  pixel positions  $\{p_1, \dots, p_S\}$ , which are obtained by downsampling the  $S$  pseudo point labels to the prediction resolution. The prediction  $\tilde{m}(p_s) \in M$  on each sampled position  $p_s$  is approximated via bilinear interpolation. Finally, the point-wise binary cross-entropy loss for each instance is formulated as:

$$L_p = - \sum_s^S l_s \log \tilde{m}(p_s) + (1 - l_s) \log(1 - \tilde{m}(p_s)) \quad (7)$$

where  $l_s$  is the assigned pseudo label at  $p_s$ .

Compared with other weakly supervised solutions like BoxInst and PointSup, where the supervision is either rough box or randomly sampled points, our PLA module benefits from the geometry information of point cloud, *e.g.*, the points are naturally distributed over the surface of an object instance. As the supervision signal is applied over these points, their receptive fields are more likely to cover the whole object, thus gaining better segmentation results.

## Graph-Based Consistency Regularization

Although the PLA can produce refined pseudo labels, incorrect labels may still exist due to two reasons. 1) System error caused by calibration noise. For example, we observe that 3D points on the edge of a target may be projected to the background area in the image plane. 2) On target surfaces with low reflectivity such as the car windshield, the lasers are more likely to penetrate the surface and hit the background area. As a result, the PLA module will assign inaccurate pseudo labels to these areas. To reduce the impact of these incorrect pseudo labels, we design GCR to further regularize the predictions of instance segmentation.

**Graph Construction.** Given the point set  $\mathcal{P}_{refine}$  generated by the PLA module, we first construct a graph  $G = \langle V, E \rangle$ , where the vertex set  $V$  is  $\mathcal{P}_{refine}$  and the edges  $E$  is weighted by the sum of image and geometry similarity as follows:

$$W_{ij} = w_1 S_{img}(i, j) + w_2 S_{geo}(i, j) \quad (8)$$

where  $w_1$  and  $w_2$  are the weighting coefficient.  $S_{img}(i, j)$  and  $S_{geo}(i, j)$  are the similarities between  $p_i$  and  $p_j$  in 2D image semantic space and 3D geometry space, respectively. For the image feature, we adopt the feature map  $F \in \mathbb{R}^{H \times W \times C}$  extracted from the CNN backbone pre-trained on the ImageNet (Deng et al. 2009). Then, the point-wise feature is obtained by performing bilinear interpolation on  $F$ , which can be denoted as  $f(p) \in \mathbb{R}^C$ . Then, the point-wise image feature similarity between  $p_i$  and  $p_j$  can be measured as follows:

$$S_{img}(i, j) = f(p_i)^T \cdot f(p_j) \quad (9)$$

Since we have the correspondence between the 3D points and the projected pixels. Thus, given the 2D points set  $\mathcal{P}_{refine} \subset \mathbb{R}^2$ , we can obtain their 3D point coordinates  $\mathcal{P}_{3d} \subset \mathbb{R}^3$ . Then, we use normalized Euclidean distance to calculate the similarity between points:

$$S_{geo}(i, j) = \exp\left(-\frac{\|p_{3d}^i - p_{3d}^j\|_2}{m}\right) + 1 \quad (10)$$

where  $m$  is the normalization constant and the  $\|\cdot\|_2$  is the  $l_2$ -norm. Then, we use a weighted sum to calculate the final similarity as in Eq. 8. Appearance feature is easily affected by occlusion and light various while geometry feature may fail due to long distance and calibration noise. Leveraging the complementary perspectives in multimodal data effectively enhances the model and gives more robust predictions.

**Consistency Regularization.** Previous studies in semi-supervised learning indicates that points on the same structure (e.g., typically referred to as a cluster or manifold) are more likely to have the same label (Zhou et al. 2003). To this end, we regularize the points with high similarity measured by Eq. 8 to share the same label.

Specifically, we first define a threshold  $\tau$ , where the edges with similarity above  $\tau$  will be set to 1 and otherwise 0:

$$e_{ij} = \begin{cases} 1 & , \quad \text{if } W_{ij} > \tau \\ 0 & , \quad \text{otherwise} \end{cases} \quad (11)$$

where  $e_{ij} \in E$ . In this way, the edge of graph  $G$  represents the label consistency between two nodes. Then, we attempt to enforce the network to yield consistent predictions according to graph  $G$ .

Let  $\tilde{m}(p) \in (0, 1)$  denote the prediction value at the position  $p$ , the graph-based regularization means that if the edge  $e_{ij}$  is 1, the prediction between  $\tilde{m}(p_i)$  and  $\tilde{m}(p_j)$  should be as close as possible. Then, the consistency loss can be instantiated a cross-entropy loss:

$$L_g = -\frac{1}{N} \sum_{i=0}^N \sum_{j=0}^N e_{ij} \log P(\tilde{m}(p_i) = \tilde{m}(p_j)) \quad (12)$$

where  $N = |V|$  and  $P(\cdot)$  is the probability.  $P(\tilde{m}(p_i) = \tilde{m}(p_j))$  is given by  $\tilde{m}(p_i) \cdot \tilde{m}(p_j) + (1 - \tilde{m}(p_i)) \cdot (1 - \tilde{m}(p_j))$  to describe the consistency. We find that the consistency loss can enforce semantically similar points to have the same predictions, leading to smoother segmentation results.

The final loss is then formulated as the combination of the two losses, i.e.,  $L = L_p + L_g$ , where  $L_g$  is the consistency regularization loss and  $L_p$  indicates the point-wise LiDAR loss in the PLA module.

## Experiments

### The nuInsSeg Dataset

To ensure an accurate and fair evaluation for our *LWSIS* and existing instance segmentation models under autonomous driving scenarios, 2D instance segmentation annotations are required and they should be consistent with the 3D bounding box annotations. To this end, we build the instance segmentation annotation for the nuScenes (Caesar et al. 2020) dataset and named it *nuInsSeg*. To our best knowledge, *nuInsSeg* is the first dataset that jointly contains LiDAR point cloud, RGB images, 3D bounding box, 2D bounding box and manually annotated instance mask that is consistent with the 2D and 3D box annotations. The comparison analysis with existing datasets is shown in Table 2.

Next, we show the properties of *nuInsSeg* dataset. The training set contains 789,193 instance mask annotations aligned to 3D bounding box annotations over 168,780 images and the validation set has 157,879 mask annotations over 36,114 images. The annotations cover 10 classes, which is the same as nuScenes (Caesar et al. 2020) 3D detection dataset. The number of instances and annotated pixels are shown in Fig 4. By contrast, *nuInsSeg* has more instance categories than Waymo (Sun et al. 2020) dataset and has a quantitative advantage over nuImage (Caesar et al. 2020) dataset.

### Implementation Details

We conduct our experiment on the *nuInsSeg* and Waymo datasets. We adopt a standard evaluation metric (He et al. 2017) of instance segmentation. It includes AP, AP<sub>50</sub>, AP<sub>75</sub> and AP<sub>s</sub>, AP<sub>m</sub>, AP<sub>l</sub> (AP at different scale). In our experiment, Mask R-CNN (He et al. 2017) and CondInst (Tian, Shen, and Chen 2020) are implemented using the official codebase without modification. We train the model for 90K iterations with batch size 16 on 4 NVIDIA Tesla V100 GPUs

Supervision	Model	Backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>s</sub>	AP <sub>m</sub>	AP <sub>l</sub>
Fully Sup.	Mask R-CNN(He et al. 2017)	ResNet50	47.55	68.24	51.98	18.14	45.35	64.88
		ResNet101	49.14	69.99	53.92	18.83	46.80	66.91
	CondInst(Tian, Shen, and Chen 2020)	ResNet50	44.88	67.17	47.53	14.01	42.85	62.26
		ResNet101	46.88	68.23	50.36	14.49	45.47	67.44
Weakly Sup.	PointSup(Cheng, Parkhi, and Kirillov 2022) LWSIS+PointSup	ResNet50	43.80	66.05	46.62	16.74	41.62	59.57
			<b>45.46(+1.7)</b>	66.74	49.01	18.10	43.76	60.70
	PointSup(Cheng, Parkhi, and Kirillov 2022) LWSIS+PointSup	ResNet101	44.72	66.15	48.17	16.72	42.69	60.73
			<b>46.17(+1.4)</b>	67.75	49.92	18.96	44.27	61.36
	BoxInst(Tian et al. 2021) LWSIS+BoxInst	ResNet50	33.40	62.61	32.03	11.08	31.98	50.21
			<b>35.65(+2.3)</b>	64.24	35.64	11.41	33.75	53.75
BoxInst(Tian et al. 2021) LWSIS+BoxInst	ResNet101	34.18	63.87	32.55	11.43	32.69	49.98	
		<b>36.22(+2.0)</b>	65.22	36.27	11.49	34.05	54.64	

Table 1: Compared with state-of-the-art methods on *nulnsSeg* val dataset. Fully Sup. means methods with fully mask supervision while Weakly Sup. means using weakly supervision. The PointSup (Cheng, Parkhi, and Kirillov 2022) uses 10 annotated points and bounding box for each instance as supervision while BoxInst (Tian et al. 2021) only use bounding box.

Dataset	Inst.	Annotation			Coh.
		2D Box	3D Box	Mask	
KITTI	80K	✓	✓	-	-
ApolloScape	-	-	✓	✓	-
A2D2	43K	✓	✓	-	-
DVPS	-	-	-	*	✓
KITTI-360	68K	✓	✓	*	✓
Waymo	12M	✓	✓	✓	-
ONCE	417K	-	✓	-	-
<b>nulnsSeg</b>	947K	✓	✓	✓	✓

Table 2: Comparison with other autonomous driving datasets. ‘\*’ means these instance masks are not manually annotated. ‘Coh.’ means if the 3D annotations are consistent with the 2D ones for each instance.

Supervision	Model	Backbone	AP
Fully Sup.	Mask R-CNN	ResNet50	43.78
	CondInst	ResNet50	43.07
Weakly Sup.	BoxInst	ResNet50	34.53
	LWSIS+BoxInst	ResNet50	<b>37.77(+3.2)</b>

Table 3: Performance of *LWSIS* on Waymo val dataset.

for both datasets. We adopt ResNet-50 and ResNet-101 pre-trained on ImageNet (Deng et al. 2009) as backbones.

## Experimental Results

**Main Results** Our method is compared with competitive fully and weakly supervised instance segmentation methods on *nulnsSeg* and Waymo (Sun et al. 2020) dataset. As shown in Table 1, with the same ResNet-50-FPN backbone, our *LWSIS* applied to BoxInst can improve the baseline by 2.2% mask AP and 3.5% AP<sub>l</sub>. With ResNet-FPN-101 backbone, *LWSIS* can improve the baseline by 2.0% and 4.6% AP<sub>l</sub>. Also, with stronger backbone, the performance of *LWSIS* can be further improved. When we adapt to the setting of PointSup, which utilizes several points as the supervision, our *LWSIS* can also achieve much better performance. It is well worthy mentioning that our weakly supervised model

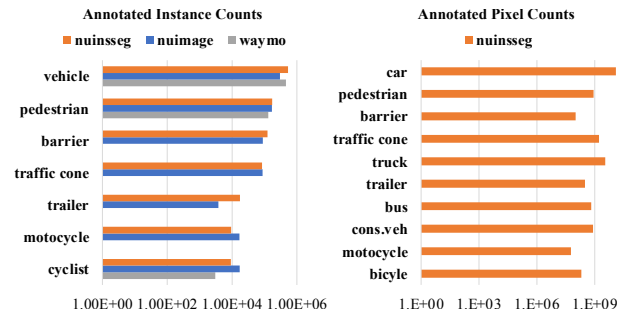


Figure 4: (a) Number of annotated instances for different datasets. (b) Number of finely annotated pixels.

based on PointSup even outperforms the fully supervised approach CondInst by 0.6% AP (45.46 vs 44.88 mAP). We also achieve about 96% of a superior fully supervised method MaskRCNN (45.46 vs 47.55 mAP).

To demonstrate the generalizability of our method, we also conduct experiment on Waymo open dataset (Sun et al. 2020). More recently, Waymo has released 2D instance segmentation labels for 692 sequences containing 61,475 images with segmentation labels. Since it has no correspondence between 2D and 3D instance labels, an IoU-based matching strategy is performed to match our setting. As shown in Table 3, our method still improves the BoxInst (Tian et al. 2021) by 3.2% mAP on Waymo dataset.

**Scale-Up for Weakly Supervised Training.** Our method can leverage more training data to improve the performance of 2D instance segmentation at a low annotation cost, and even outperforming the fully supervised models. For simplicity, we use a certain proportion of the data with mask annotations from the *nulnsSeg* training set. The ‘Costs’ is measured by annotation time according to (Cheng, Parkhi, and Kirillov 2022) which estimates the average manually annotating time. They found that it takes 0.9, 7, 79.2 seconds to label point, bounding box and polygon-based instance mask respectively.

The blue line in Fig. 5 shows the mAP of Mask R-

Methods	mAP	Car	Truck	Bus	Trailer	Ctr.	Ped.	Motor.	Bicycle	Tr.Cone	Barrier
CenterPoint	56.7	85.1	54.0	65.8	35.6	14.3	84.3	55.1	39.0	67.9	66.3
Improved CenterPoint	62.2	86.3	58.5	65.6	38.4	19.1	86.7	66.8	55.7	76.3	68.7
Delta	+5.5	1.2	4.5	-0.2	2.8	4.8	2.4	11.7	6.7	8.4	2.4

Table 4: 3D detection performance on *nuScenes* val dataset. Abbreviations: construction vehicle(Ctr.), pedestrian(Ped.) and traffic cone(Tr.Cone). We improve CenterPoint with the 2D segmentation results predicted by our method.

Model	PLA	GCR		AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>s</sub>	AP <sub>m</sub>	AP <sub>l</sub>
		Image	Geometry						
BoxInst	-	-	-	33.40	62.61	32.03	11.08	31.98	50.21
<b>Ours</b>	✓	-	-	34.77	63.85	33.90	10.28	31.53	<b>55.43(+5.2)</b>
	-	✓	-	34.32	63.72	32.95	11.58	32.55	50.67
	-	-	✓	34.15	63.30	32.85	<b>11.93(+0.9)</b>	32.67	49.97
	✓	✓	-	35.27	63.98	35.03	11.40	33.08	53.19
	✓	✓	✓	<b>35.65(+2.3)</b>	<b>64.24(+1.6)</b>	<b>35.64(+3.6)</b>	11.41	<b>33.75(+1.8)</b>	53.75

Table 5: Ablation studies of *LWSIS* by verifying each module on *nuInsSeg* val dataset.

CNN (He et al. 2017) vs. annotation costs of different proportion of training data of *nuInsSeg*. Under the same annotation costs, *LWSIS* based on PointSup method can achieve 10.8% mAP improvement over the fully supervised MASK R-CNN. This indicates that our method will reduce the annotation costs by 60% to achieve same mAP (e.g., Mask R-CNN needs 2.5 times the annotation cost of our model to achieve 45.4 mAP).

**Improving Downstream Tasks** In Table 4, we show an example that our *LWSIS* improves the performance of 3D object detector. The experiment basically follows the implementation of PointPainting (Vora et al. 2020), which requires a 2D segmentation network and a 3D object detector. In particular, CenterPoint (Yin, Zhou, and Krahenbuhl 2021) with VoxelNet (Yan, Mao, and Li 2018) backbone are adopted as the baseline 3D detector, while *LWSIS* based on BoxInst are used as the 2D segmentation network. First, the images are passed through *LWSIS* to obtain pixel-wise category label. Then, the LiDAR points are projected onto the images to get the pixel-wise category results. Finally, the point cloud with category label are fed to 3D detector to obtain improved 3D detections. This shows that with only LiDAR point cloud and 3D box annotations, our method can further improve the performance of 3D object detectors.

### Ablation Study

Table 5 demonstrates the performance improvement of the PLA module and the GCR module, where we choose BoxInst as our baseline. With the PLA module alone, the performance is improved by 1.4% AP and by near 5.2% AP<sub>l</sub>, which shows that the supervision information produced by the PLA module is especially effective for objects of large scale. The two components of GCR module are better for small and medium objects which proves that GCR module can provide more accurate supervision with consistency regularization when the point cloud is sparse for distant object. Also, mask AP can be further improved by 2.3% when PLA and GCR are applied together which indicate that they can

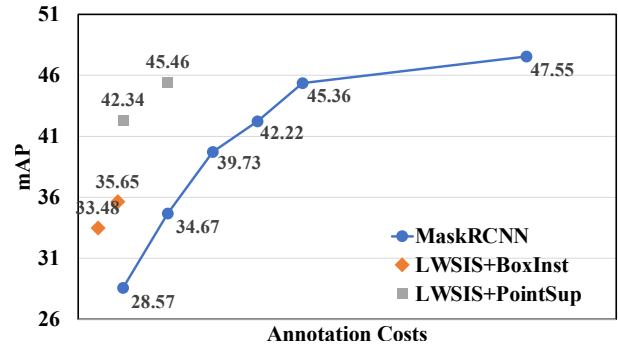


Figure 5: The mAP vs data labeling costs for different methods on *nuInsSeg* val dataset. The x-axis is the annotation time for 10% ~ 100% of the *nuInsSeg* train dataset.

provide complementary supervision information. The PLA module takes advantage of pseudo point annotations from LiDAR, while GCR explores the similarity between points to correct some inaccurate point labels.

### Conclusions

This paper proposed a new multimodal weakly supervised instance segmentation framework, named *LWSIS*, which tactfully exploits the 3D LiDAR point cloud as supervision of 2D images. The core idea is to fully utilize the geometry information of the point cloud to guide the training of 2D instance segmentation models. The obtained 2D segmentation models can also in turn help the learning of 3D perception tasks by multimodal fusion. Our *LWSIS* is composed of two crucial modules: PLA module and GCR module, which explicitly assign the point-level annotations and implicitly regularize the inaccurate predictions, respectively. Furthermore, our *nuInsSeg* contributed instance segmentation annotations based on nuScenes to the community. We expect our work could encourage further research on multimodal perception in the context of autonomous driving.

## Acknowledgments

The research was supported by the Start-up Research Grant (SRG) of University of Macau (SRG2022-00023-IOTSC), FDCT grant SKL-IOTSC(UM)-2021-2023 and the Science and Technology Commission of Shanghai Municipality (grant No. 22DZ1100102). We would like to thank the anonymous referee for valuable suggestions that helped us to significantly improve this paper.

## References

- Ahn, J.; Cho, S.; and Kwak, S. 2019. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2209–2218.
- Arun, A.; Jawahar, C. V.; and Kumar, M. P. 2020. Weakly Supervised Instance Segmentation by Learning Annotation Consistent Instances. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *ECCV*.
- Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2016. What’s the point: Semantic segmentation with point supervision. In *ECCV*.
- Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*.
- Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; and Yan, Y. 2020. Blendmask: Top-down meets bottom-up for instance segmentation. In *CVPR*.
- Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. 2019. Hybrid task cascade for instance segmentation. In *CVPR*.
- Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4): 834–848.
- Cheng, B.; Parkhi, O.; and Kirillov, A. 2022. Pointly-Supervised Instance Segmentation. In *CVPR*, 2617–2626.
- Cholakkal, H.; Sun, G.; Khan, F. S.; and Shao, L. 2019. Object Counting and Instance Segmentation With Image-Level Supervision. In *CVPR*.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.
- Ge, W.; Guo, S.; Huang, W.; and Scott, M. R. 2019. LabelPEnet: Sequential Label Propagation and Enhancement Networks for Weakly Supervised Instance Segmentation. In *ICCV*.
- Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*.
- Hsu, C.-C.; Hsu, K.-J.; Tsai, C.-C.; Lin, Y.-Y.; and Chuang, Y.-Y. 2019. Weakly Supervised Instance Segmentation using the Bounding Box Tightness Prior. In *NeurIPS*.
- Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M.; and Schiele, B. 2017. Simple Does It: Weakly Supervised Instance and Semantic Segmentation. In *CVPR*.
- Kirillov, A.; Wu, Y.; He, K.; and Girshick, R. 2020. Pointrend: Image segmentation as rendering. In *CVPR*.
- Lan, S.; Yu, Z.; Choy, C.; Radhakrishnan, S.; Liu, G.; Zhu, Y.; Davis, L. S.; and Anandkumar, A. 2021. DISCOBOX: Weakly Supervised Instance Segmentation and Semantic Correspondence from Box Supervision. In *ICCV*.
- Lee, J.; Yi, J.; Shin, C.; and Yoon, S. 2021. BBAM: Bounding Box Attribution Map for Weakly Supervised Semantic and Instance Segmentation. In *CVPR*.
- Lee, J.-H.; Kim, C.; and Sull, S. 2021. Weakly Supervised Segmentation of Small Buildings With Point Labels. In *ICCV*, 7406–7415.
- Meng, Q.; Wang, W.; Zhou, T.; Shen, J.; Gool, L. V.; and Dai, D. 2020. Weakly supervised 3d object detection from lidar point cloud. In *ECCV*, 515–531. Springer.
- Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*.
- Song, C.; Huang, Y.; Ouyang, W.; and Wang, L. 2019. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *CVPR*.
- Sun, P.; Kretschmar, 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 *CVPR*.
- Tang, M.; Djelouah, A.; Perazzi, F.; Boykov, Y.; and Schroers, C. 2018. Normalized Cut Loss for Weakly-Supervised CNN Segmentation. In *CVPR*.
- Tian, Z.; Shen, C.; and Chen, H. 2020. Conditional convolutions for instance segmentation. In *ECCV*.
- Tian, Z.; Shen, C.; Wang, X.; and Chen, H. 2021. Boxinst: High-performance instance segmentation with box annotations. In *CVPR*.
- Vora, S.; Lang, A. H.; Helou, B.; and Beijbom, O. 2020. Pointpainting: Sequential fusion for 3d object detection. In *CVPR*.
- Wang, X.; Feng, J.; Hu, B.; Ding, Q.; Ran, L.; Chen, X.; and Liu, W. 2021. Weakly-Supervised Instance Segmentation via Class-Agnostic Learning With Salient Images. In *CVPR*.
- Wang, Y.; Yin, J.; Li, W.; Frossard, P.; Yang, R.; and Shen, J. 2023. SSDA3D: Semi-supervised Domain Adaptation for 3D Object Detection from Point Cloud. In *AAAI*. AAAI Press.
- Xu, L.; Ouyang, W.; Bennamoun, M.; Boussaid, F.; Sohel, F.; and Xu, D. 2021. Leveraging Auxiliary Tasks With Affinity Learning for Weakly Supervised Semantic Segmentation. In *ICCV*, 6984–6993.
- Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.
- Yin, J.; Fang, J.; Zhou, D.; Zhang, L.; Xu, C.-Z.; Shen, J.; and Wang, W. 2022a. Semi-supervised 3D object detection with proficient teachers. In *ECCV*, 727–743. Springer.



- Yin, J.; Shen, J.; Gao, X.; Crandall, D.; and Yang, R. 2021. Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds. *TPAMI*.
- Yin, J.; Zhou, D.; Zhang, L.; Fang, J.; Xu, C.-Z.; Shen, J.; and Wang, W. 2022b. Proposalcontrast: Unsupervised pre-training for lidar-based 3D object detection. In *ECCV*, 17–33. Springer.
- Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *CVPR*.
- Zhang, G.; Lu, X.; Tan, J.; Li, J.; Zhang, Z.; Li, Q.; and Hu, X. 2021. Refinemask: Towards high-quality instance segmentation with fine-grained features. In *CVPR*.
- Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; and Schölkopf, B. 2003. Learning with Local and Global Consistency. In *NIPS*.