Beyond the Label Itself: Latent Labels Enhance Semi-supervised Point Cloud Panoptic Segmentation

Yujun Chen1,2,*, Xin Tan1,2,*, Zhizhong Zhang1,2,†, Yanyun Qu3, Yuan Xie1,2

1School of Computer Science and Technology, East China Normal University
2Chongqing Institute, East China Normal University
3School of Information Science and Engineering, Xiamen University
51215901065@stu.ecnu.edu.cn, {xtan, zzzhang, yxie}@cs.ecnu.edu.cn, yyqu@xmu.edu.cn

Abstract
As the exorbitant expense of labeling autopilot datasets and the growing trend of utilizing unlabeled data, semi-supervised segmentation on point clouds becomes increasingly imperative. Intuitively, finding out more “unspoken words” (i.e., latent instance information) beyond the label itself should be helpful to improve performance. In this paper, we discover two types of latent labels behind the displayed label embedded in LiDAR and image data. First, in the LiDAR Branch, we propose a novel augmentation, Cylinder-Mix, which is able to augment more yet reliable samples for training. Second, in the Image Branch, we propose the Instance Position-scale Learning (IPSL) Module to learn and fuse the information of instance position and scale, which is from a 2D pre-trained detector and a type of latent label obtained from 3D to 2D projection. Finally, the two latent labels are embedded into the multi-modal panoptic segmentation network. The ablation of the IPSL module demonstrates its robust adaptability, and the experiments evaluated on SemanticKITTI and nuScenes demonstrate that our model outperforms the state-of-the-art method, LaserMix.

Introduction
Point clouds, with richer visual and geometric information, have played an increasingly significant role in perception tasks (Roriz, Cabral, and Gomes 2021; Triess et al. 2021). Given its extensive potential, point cloud panoptic segmentation, unifying instance and semantic segmentation, has been applied in various fields, such as autonomous driving, robotics, and industrial manufacturing (Geiger, Lenz, and Urtasun 2012; Fernandes et al. 2021; Wang and Kim 2019).

However, it is exhausting to annotate point cloud data, which prohibitively restricts its potential applications (Hu et al. 2022; Unal, Dai, and Van Gool 2022; Xu and Lee 2020). Hence, it is highly demanded to utilize less point data to achieve better performance. Nowadays, images play a crucial role in point cloud segmentation since they are cost-effective (Cui et al. 2021; Li and Lee 2021). In addition, with the larger number of unlabeled point cloud data collected, the trend towards semi-supervision is prevalent (Sindhwani, Niyogi, and Belkin 2005; You et al. 2022). Thus, in this paper, we pay attention to semi-supervised multi-modal point cloud (panoptic) segmentation (SMPS).

We have observed two major limitations in the previous SMPS methods. First, due to the limited ground-truth data, the preceding semi-supervised (SS) models generated pseudo-labels as ground-truth for training (Kong et al. 2023; Park et al. 2022). Nevertheless, such pseudo-labels are sometimes unreliable and of poor quality when the network is not fully trained. Second, for the 3D-2D cross-modal processing, previous multi-modal point cloud models (Liu et al. 2022; Liang et al. 2022) treat all pixels equally without leveraging sufficient image-level information (e.g., size and boundary), despite the fact that images have a superior ability to detect and locate objects, which even outperforms point clouds in certain cases (Park et al. 2022).

According to the above discussions, we argue that the information contained in point cloud labels is not only as superficial as the displayed label itself but also encompasses some latent labels beneath the surface. Hence, it is a feasible solution to explore unspoken words (i.e., latent labels) to address the two issues. Specifically, for the first, since only given labeled data is reliable and completely correct, 1) can we exploit self-information among labeled point clouds to construct other reliable and diverse labels? For the second
one, the existing 3D instance labels can be projected onto the image as a form of weak 2D annotations, which indicate the instances’ position and scale. Therefore, 2) can we utilize these weak annotations from 3D-2D self-information to improve the network segmentation performance?

In this paper, we propose to address the limitations by discovering more latent labels. As illustrated in Figure 1, our method significantly surpasses previous semi-supervised point cloud segmentation ones.

First, we propose a novel data augmentation method, Cylinder-Mix, to obtain reliable and diverse samples for training. Specifically, to exploit self-information within limited data, it performs interleaved mixing for labeled point clouds based on cylinder voxels. This kind of mix strategy does not require other annotations, and the obtained labels are as accurate as the human-annotated ones.

Second, Instance Position-Scale Learning (IPS) Module is proposed to learn the instance information of position and boundary on images. In this module, latent labels from 3D-2D self-information, the instance boxes, are obtained through LiDAR-to-camera projection. Subsequently, inspired by interactive image segmentation (IIS) (Ramadan, Lachqar, and Tairi 2020; Lin et al. 2020), weak interactive annotation (e.g., the bounding box with location and boundary information) leads the network to pay more attention to the highlighted object. Hence, the boxes, which serve as interactive annotations, are encoded into heatmaps to indicate positions and scales, and then fused to the 2D backbone. This way, these latent labels from 3D-2D projection highlight the instances so as to improve segmentation.

Furthermore, our experiments demonstrate the adaptability and versatility of the IPSL module, as it still works well whether it fuses instance boxes from a trained detector or masks from a large model (e.g., SAM (Kirillov et al. 2023), and Grounded-Segment-Anything (Contributors 2023)). Additionally, to our best knowledge, our model is the first multi-modal model applying the semi-supervised setting to solve point cloud panoptic segmentation.

In summary, our contributions are summarized as follows:

- We explore latent labels from self-information on both point clouds and images, without requiring any extra labels, to enhance network segmentation performance. Surprisingly, our model finally achieves state-of-the-art performance on semi-supervised semantic segmentation, surpassing the previous best method, LaserMix.
- We propose a novel data augmentation technique, Cylinder-Mix, that could obtain reliable and diverse labels, as accurate as the given ground-truth, within limited labeled data for semi-supervised training.
- We propose the Instance Position-scale Learning (IPS) Module to fuse latent annotations from 3D-2D self-information. The obtained latent labels, instance boxes, are utilized to provide position and scale explicitly.

### Related Work

#### Fully-supervised Point Cloud Segmentation

In recent years, the efficiency of point cloud segmentation has been enhanced mainly by 3D perspective feature extraction (Zhou, Zhang, and Foroosh 2021; Li et al. 2022a; Fong et al. 2022; Qi et al. 2017) and multi-modal fusion (Liu et al. 2022; Liang et al. 2022; Zhang et al. 2023; Tan et al. 2023; Liu et al. 2021). Cylinder-3D (Zhu et al. 2021) proposes a novel voxelization approach for the geometric properties of the point cloud, while Panoptic-Polarnet (Zhou, Zhang, and Foroosh 2021) utilizes the Bird’s Eye View (BEV) feature for segmentation and clustering, which not only realizes the panoramic segmentation, but also avoids the occlusion issue between instances. However, the 2D image branch has been somewhat neglected, leaving the network far from better performance during semi-supervised tasks. Although BEFusion (Liu et al. 2022) combines images, it treats all pixels equally without using instance information. Distinctively, we do not only fuse the image features but also focus more on the instance position and boundary of the image, so as to implicitly enhance the performance of the SS segmentation network in the process of multi-modal fusion.

#### Semi-Supervised Point Cloud Methods

Several methods for indoor scenes have been proposed. SSPC-Net (Cheng et al. 2021) utilizes pseudo-label propagation through superpixel segmentation, while Jiang et al. (2021) applied Guided Point Contrastive Loss to semi-supervised indoor datasets. However, such methods are unable to effectively handle large-scale real-world datasets, such as SemanticKITTI and nuScenes, due to their large number of points and objects. For methods on real-world datasets, LaserMix (Kong et al. 2023), adapting Mean Teacher (Tarvainen and Valpola 2017) architecture to SS point segmentation, proposes a new circular mixture data augmentation. However, the generated samples from LaserMix can be unreliable due to the mixture of unlabeled data since the network is not always well-trained. DetMatch (Park et al. 2022) applies the Mean Teacher architecture to detection tasks, but lacks sufficient adaptability for panoptic segmentation like Panoptic-PolarNet. In contrast, our proposed data augmentation, Cylinder-Mix, not only generates reliable and diverse labels from limited labeled data for training but also inherits the excellent feature extraction capability of Cylinder3D, thus making it adaptable for semi-supervised panoptic segmentation tasks on the real-world dataset.

#### Preliminaries

##### Problem Setting

Semi-supervision usually uses the whole points of partial samples or frames. We adopt a sampling method of selecting frames with fixed intervals, which is the same as Kong et al. (2023). In this paper, we have respectively selected 40%, 20%, 10%, and 1% point cloud frames with point-wise labels for training.

#### Symbols

We use $P = \{ (x, y, z, b) \}_{n=1}^{N}$ to denote a 3D LiDAR point cloud, with the RGB image(s) $I = \{ I \in \mathbb{R}^{H \times W \times 3} \}_{1}$, where $(x, y, z)$ indicates the spatial coordinates, $b$ is the brightness of each point, $N$ and $v$ are respectively the number of points and camera views. Besides, there are point-wise semantic and panoptic labels $L = \{ l_{sem}, l_{nst} \}_{n=1}^{N}$, $l_{sem} \in [1, 2, 3, ..., C]$, where $C$ is the number of semantic categories. Therefore, the dataset could be represented as $P = \{ P, I, L \}_{1}^{M}$ with $M$ frames. During training, the dataset is divided into labeled and unlabeled.
Cylinder-Mix Augmentation

SwiftNet

Image feature \( F_i \)

Camera RGB Images

LiDAR Point Cloud

Cylinder-Mix Augmentation

LiDAR Branch

Multi-modal Segmentation Network

Multi-modal BEV features

Panoptic-PolarNet

\( F_{\text{in}} \)

\( F_{\text{BEV}} \)

Semantics

Fore-mask

Images and instance boxes

3D-2D projection

IIS fusion

BEV pooling

Camera RGB Images

SwiftNet

Image feature \( F_i \)

Figure 2: The framework of our model. Our model is composed of three parts. LiDAR Branch, on the 3D point cloud branch, gets better 3D features through self-supervised augmentation, called Cylinder-Mix, while Image Branch improves the 2D backbone via fusion of instance position and scale information. After that, both cross-modal features will be fused to the BEV feature, following Multi-modal Segmentation Network to extract features and get point-wise labels in the end.

parts, denoted by the superscript \( l \) and \( u \), respectively. For example, \( P^u \) means unlabeled points of the dataset.

**Method**

As shown in Figure 2, LiDAR Branch, given the labeled point clouds, outputs point-wise features through Cylinder-Mix augmentation and Cylinder3D (Zhu et al. 2021) extractor network. On Image Branch, the proposed Instance Position-Scale Learning (IPSL) module outputs instance heatmaps. To this end, a detector, supervised by the 3D-2D multi-modal fusion and adapts well to the feature extraction of Cylinder3D. It employs Cylinder-Mix for achieving more latent samples and self-supervised enhancement.

As shown in the blue part of Figure 2, LiDAR Branch learns point-wise features with the input of a point cloud \( P \). Firstly, it voxelizes the points into cylindrical voxels with the grid size of \( [G_x, G_y, G_z] \) where \( G_x \) is the number of blocks of voxels to be split along the X-axis, so as to \( G_y, G_z \). This way we get the grid index \( \{v_x, v_y, v_z\} \) that denotes which cylinder-voxel the point \( (x, y, z) \) belongs to. Secondly, it applies our proposed Cylinder-Mix to increase the diversity of the point clouds within the limited data.

The augmented points \( P_{mix} \) are fed into the MLP-based feature extractor of Cylinder3D, thus getting point-wise features \( F_{pt} \in \mathbb{R}^{N \times C_b} \). Finally, through the grid index of points and BEV pooling, we aggregate \( F_{pt} \) into LiDAR BEV features \( F_{BEV}^{(PT)} \) with the size of \( G_x \times G_y \times G_z \times C_b \), where \( C_b \) is the number of channels of the BEV feature. When pooling the voxel at position \( (v_x, v_y, v_z) \), its BEV feature is the maximum of point features \( F_{pt}(p) \) within the point set \( S_{(v_x, v_y, v_z)} \) in which all points belong to this voxel.

\[
F_{BEV}^{(PT)}(v_x, v_y, v_z) = \max_{p \in S_{(v_x, v_y, v_z)}} F_{pt}(p). \tag{1}
\]

### Cylinder-Mix Augmentation

To obtain more data for network training, we propose a point cloud mixture data augmentation, Cylinder-Mix. It, without additional annotations, provides more diverse point clouds for network training and adapts well to the feature extraction of Cylinder3D. To fully mix the point cloud samples, we perform an interleaved mixing as figure 3. This means that for a certain block region of the mixed point cloud, its surrounding regions (i.e., above, below, to the left, and to the right) all come from another different point cloud.

Given the point cloud samples \( P_1, P_2 \) and their cylinder voxel index of each point \( V = \{(v_x, v_y, v_z)\}_{n=1}^N \) with the grid size of \( [G_x, G_y, G_z] \), we first divide the two point clouds into mixture regions with the size \( [R_x, R_y, R_z] \) along each \( x, y \) and \( z \) axis. For example, the region size of the mixture shown in Figure 3 is \( [8, 1, 2] \). To mix them, we calculate the \((r_x, r_y, r_z)\) of each point, where \( r_x \in \{0, 1, 2, ..., R_x - 1\} \) represents the index number of the mixture region, which point belongs to the \( x \) axis, similarly for \( r_y \) and \( r_z \).

\[
r_k = \left\lfloor \frac{v_k}{G_k \times R_k} \right\rfloor, \quad (k = x, y, z). \tag{2}
\]

For interlaced mixing, the two point clouds \( P_1, P_2 \) are mixed according to index-judgment as the following:

\[
\delta(r_k) = \begin{cases} True, & \text{if } mod(r_k, 2) = 0 \\ False, & \text{if } mod(r_k, 2) = 1 \end{cases}, \quad (k = x, y, z). \tag{3}
\]
Another to project the point through camera intrinsic parameters and vehicle parameters, BEV features, we employ the LiDAR-to-camera projection, encodes initial pixel-wise features the instance information from the IPSL Module, SwiftNet corresponding images of the point cloud, this branch is ex-

This module will be presented in the next section. Given the Position-scale Learning (IPSL) Module. Further details on consists mainly of two parts: the backbone network (Swift-

Net (Orsic et al. 2019) in our experiment) and the Instance Pipeline Image Branch is aimed at extracting image fea-

tures to get the camera BEV feature. Its value on \( v \) = \((x, y, z)\) is a judgment function for judging whether the point at \((x, y, z)\) will be mixed into the point clouds \( P_{mix1} \). To get the another mixed point clouds \( P_{mix2} \), the index-judgment function \( J_2 = \neg J_1 \).

\[
J(x, y, z) = \neg (\delta (r_x) \oplus \delta (r_y) \oplus \delta (r_z)). \tag{4}
\]

In equation 4, \( \neg \) and \( \oplus \) indicate logical negation and xor, respectively. \( J_1(x, y, z) \) is a judgment function for judging whether the point at \((x, y, z)\) will be mixed into the point clouds \( P_{mix1} \). To get the another mixed point clouds \( P_{mix2} \), the index-judgment function \( J_2 = \neg J_1 \).

\[
P_{mix2} = P_1[J_1(P_1)] \cup P_2[J_1(P_2)]. \tag{5}
\]

On this basis, we get mixed points by equation 5, as shown in Figure 3. Attention that for the mixed point cloud from \( P_1 \) and \( P_2 \), its Image Branch should perform feature extraction and projection for camera images \( I_1 \) and \( I_2 \), respectively. In addition, consistently training with mixed data can potentially cause the model distribution to deviate from that of training with original data. Hence, we set \( p_{polymix} \), the probability of using Cylinder-Mix to control this augmentation.

Training Image Branch

Pipeline Image Branch is aimed at extracting image features and learning instance information from some weak labels. As shown in the orange part of Figure 2, Image Branch consists mainly of two parts: the backbone network (SwiftNet (Orsic et al. 2019) in our experiment) and the Instance Position-scale Learning (IPSL) Module. Further details on this module will be presented in the next section. Given the corresponding images of the point cloud, this branch is expected to output camera BEV features.

With the input images \( I \) of point clouds and the fusion of the instance information from the IPSL Module, SwiftNet encodes initial pixel-wise features \( F_{im} \). To obtain camera BEV features, we employ the LiDAR-to-camera projection, through camera intrinsic parameters and vehicle parameters, to project the point \( p = (x, y, z) \) onto the camera coordinate system \( q = (h, w) \). We obtain a set of matched pairs mapping \( M = \{p, q\} \) between 3D points and 2D pixels. Subsequently, we perform BEV pooling on the matched pixel features to get the camera BEV feature. Its value on \((v_x, v_y, v_z)\) is the maximum of pixel features \( F_{im}(q) \), where \( q \) is in set \( M_{(v_x, v_y, v_z)} \) and the matched point belongs to its voxel.

\[
F_{im}^{(BEV)}(v_x, v_y, v_z) = \max_{(p, q) \in M_{(v_x, v_y, v_z)}} F_{im}(q). \tag{6}
\]

Instance Position-scale Learning Module The Instance Position-scale Learning Module is aimed to utilize weak labels on images to provide instance information for improving point cloud segmentation. Inspired by interactive image segmentation (IIS) (Sofiuk, Petrov, and Konushin 2022), images with weak annotations, such as a point indicating the center of an object, make the model pay more attention to the annotated objects. Next, we will introduce how to generate and utilize weak annotations.

Initially, to train with weak labels, it is necessary to get the latent ground-truth through the instance labels of points and LiDAR-to-camera projection. This involves matching and projecting the 3D points onto 2D pixels. For a specific object with the instance label \( y_{inst} \), its box can be enclosed by the maximum and minimum values of pixel coordinates \((h, w)\) from projected points of this instance. Subsequently, we preserve these box labels and use them as targets to train the Detection Network. The Detection Network is an offline network, and it is convenient to adopt any detection backbone. In our experiment, we directly select the detection model from MMdetection (Chen et al. 2019).

In this way, the module predicts the instance bounding boxes of cameras through the trained Detection Network. Since these boxes are in 2D coordinates, we convert them into instance heatmaps using the following equation:

\[
d_q(m, n) = \sqrt{(h - m)^2 + (w - n)^2}, \tag{7}
\]

\[
H_q(m, n) = \begin{cases} 
\exp \left(\frac{-2d_{mn}^2}{R^2}\right) & d_{mn} \leq R \\
0 & d_{mn} > R ,
\end{cases} \tag{8}
\]

where \( d_q(m, n) \) is the distance on the image coordinate system between pixel \((m, n)\) and \( q = (h, w) \). The \( H_q \) is a Gaussian-based heatmap of a box angular or box center point \( q \), with the radius of \( R \).

In implementation, the heatmap of an instance is gathered by its 4 box angular points \( q_1, q_2, q_3, q_4 \) and its center point \( q_{cen} \) of the box. \( R = 5 \) for each of the 4 corners, while for its center, \( R \) is determined by the product of the instance width and the percentage \( P_{center} \). Here, instance width is the minimum value between the box length and width. We set \( P_{center} = 1/4 \) as the default. To ascertain the optimal configuration of these parameters, we conduct comprehensive ablation experiments, as detailed in the Appendix.

With stacking the heatmaps \([H_{q_1}, H_{q_2}, H_{q_3}, H_{q_4}, H_{q_{cen}}]\) and maximizing at the stacked dimension, we get the instance heatmap of the box. Furthermore, for the camera image from view \( v \), its instances heatmap is the maximum of Gaussian heatmap of each instance.

Finally, referring to the fusion block of IIS (Sofiuk, Petrov, and Konushin 2022), these heatmaps and images along with each camera view, are fused together on intermediate features, with camera images \( I \). As shown in equation 9, images \( I \) and heatmaps \( H \) are aligned through their respective mapping heads \( \psi_H \) and \( \psi_I \), 2D convolution with the number of output channels 64. And then they are summed to obtain fused intermediate features \( F_{IH} \) and fed into the backbone of Image Branch for further extraction.

\[
F_{IH} = \psi_H(H) + \psi_I(I). \tag{9}
\]
Instance Heatmap from SAM Masks. There is another method to obtain instance heatmaps. Due to the development of large models such as CLIP (Radford et al. 2021), SAM (Kirillov et al. 2023), and Grounded-Segment-Anything (Contributors 2023), instance masks are still available even without fine-tuning autopilot camera images. Therefore, through zero-shot segmentation by Grounded-Segment-Anything, we get \( n_i \) instance masks \( Mask \) and their scores \( Sco \) on an image. The instance heatmap could be the weighted sum of all those masks.

\[
\mathcal{H} = \sum_{i=1}^{n_i} Mask_i \ast Sco_i. \tag{10}
\]

Multi-modal Segmentation Network

It is expected to get panoptic results with the backbone of Panoptic-PolarNet (Zhou, Zhang, and Foroosh 2021) and multi-heads.

\[
F^{(BEV)} = \mathcal{L}(F^{(BEV)}_{pl} \otimes F^{(BEV)}_{im}). \tag{11}
\]

The multi-modal BEV features are obtained with LiDAR and Camera ones through equation 11, where \( \otimes \) means concatenation along the channel dimension and \( \mathcal{L} \) represents the linear fully connected layer to compress the feature dimension. \( F^{(BEV)} \) is input into the Multi-modal Segmentation Network. To achieve panoramic segmentation, multi-heads include semantic and instance heads, and the latter is responsible for predicting heatmaps of offset and center. Moreover, a fore-mask head is added for \( thing \) classes perception. We utilize the above mask and heatmaps, based on predicted semantic labels, to cluster and ultimately identify instances.

Segmentation Loss. The loss of segmentation \( \mathcal{L}_{seg} \), besides semantic loss and instance losses of the panoptic network, we also added foreground mask loss. With predicted logits of each point on semantic head and ground-truth, we could get semantic loss \( \mathcal{L}_{sem} \) by cross-entropy loss. For instance head, we use MSE loss(\( \mathcal{L}_{hm} \)) and L1 Loss(\( \mathcal{L}_{os} \)) to fit the center heatmap of instances \( hm \in \mathbb{R}^{H \times W \times 1} \) and offset of each point \( os \in \mathbb{R}^{H \times W \times 2} \). For fore-mask head, we use MSE loss(\( \mathcal{L}_{fm} \)) to fit foreground mask \( fm \in \mathbb{R}^{H \times W \times 1} \), in which 1 and 0 denote \( thing \) and background classes. The final loss with coefficients is

\[
\mathcal{L}_{seg} = \mathcal{L}_{sem} + \mu_{hm}\mathcal{L}_{hm} + \mu_{os}\mathcal{L}_{os} + \mu_{fm}\mathcal{L}_{fm}. \tag{12}
\]

Self-Training

First, we train our model with labeled data and loss as equation 12, which is called pretraining. Second, pseudo labels of unlabeled data are estimated with the pretrained model. Third, after loading with the parameter of pretraining, we will retrain model with labeled and estimated data. Noted that with the decrease in the amount of labeled data during pretraining, the lower semi-supervised percentage, the more epochs are required. And the specific epoch of each setting is given in the section on Experiments.

Experiments

Dataset and Metrics

SemanticKITTI (Behley, Milioto, and Stachniss 2021) contains 10 (1/11) training (validation/testing) sequences and totally 43551 LiDAR scans with a 64-beam LiDAR sensor, as well as binocular camera images of each scan additionally. There are point-wise panoptic annotations with 8 \( thing \) class and 12 \( stuff \) class labels with instance labels.

nuScenes (Caesar et al. 2020) is a large-scale autopilot dataset with various urban scenes and a 32-beam LiDAR sensor. It totally contains 1000 driving scenes of 20s duration and point-wise panoptic annotations, with 16 semantic classes, 10 of which are \( thing \) classes. For the camera, there are 6 views of images per scan.

Metrics. Typically, panoptic quality (PQ) is used to evaluate panoptic segmentation as Kirillov et al. (2019). And mean IoU (mIoU) reflects the quality of semantic segmentation. For the detector on IPSL Module, average precision (AP) is used to measure the quality of predicted boxes.

Implementation Details

Our baseline comprises an image encoder (SwiftNet), a LiDAR encoder (Cylinder3D), and a decoder (Panoptic-PolarNet). It can be comprehended as the combination of section of Image Branch and LiDAR Branch pipeline, as well as Multi-modal Segmentation Network, with the exclusion of the description of the proposed IPSL module and Cylinder-Mix. Additionally, the instance augmentation of PUPS (Su et al. 2023) is applied.

Implementation Default settings unless ablation. In IPSL Module, the radius of Gaussian heatmaps \( R = 5 \), percentage of scale to the center \( P_{center} = 1/4 \). For Cylinder-Mix, the region size \( [R_x, R_y, R_z] = [4, 4, 2] \), the probability \( p_{d gimix} = 25\% \). For the weight of each term in loss in \( \mathcal{L}_{seg}, \mu_{hm} = 100, \mu_{os} = 10, \mu_{fm} = 1 \). More Implementation Details of Detection Network and point cloud segmentation network are in the Appendix (Chen et al. 2023).

Experimental Results

Results of Detection Network  We utilize Faster R-CNN with ResNet-101 (He et al. 2016) to train the detector. The result mAP on the fully nuScenes dataset is only 31.2%. However, these predicted instance boxes, acting as weak labels, can enhance point segmentation even without high precision. See the detailed class-wise mIoU in the Appendix.
Results of Panoptic Segmentation  

Firstly, we present the Quantitative Results as Table 1. It shows segmentation results on the validation set. As there are limited panoramic segmentation methods available for SS methods, we compare our model with previous methods (French et al. 2019; Tarvainen and Valpola 2017; Zou et al. 2018; Chen et al. 2021; Kong et al. 2023) with semantic segmentation metrics (mIoU, %) and attach our panoptic segmentation metrics (PQ, %). Eventually, our model outperforms the previous top, LaserMix, and attains state-of-the-art semantic results.

Our model excels in the semi-supervised setting, as it exhibits superior performance when given a smaller amount of labeled data. On nuScenes and SemanticKITTI at 10%, the model achieves 3.1% and 2.9% improvement on PQ and 3.3% and 3.0% improvement on mIoU, respectively, compared to the baseline. Moreover, at the setting of 20% labeled data on SemanticKITTI, our model achieves comparable performance to the fully-supervised baseline, and even surpasses that at the setting of 40%.

Visualization. Our model excels in excels in scale and position guidance. As shown in Figure 5), with the instance heatmaps, our model achieves superior segmentation in terms of instance scale. Moreover, the heatmap position guidance helps alleviate false positives in some small and distant instances. Figure 6 illustrates our model’s superior segmentation on instance boundary points. In terms of metrics, the bar graph on the right (representing points distant from the instance center) of each class shows the greatest accuracy increase, suggesting that our model pays more attention to points near the boundaries.

Ablation Studies

Compare with LaserMix  For a fair comparison, we compare our model with LaserMix under the same baseline. As demonstrated in Table 2, there are two scenarios: firstly, using LaserMix not only its framework (Mean Teacher) but also its data augmentation simultaneously (baseline + LaserMix), and secondly, employing only the augmentation while keeping other conditions constant (baseline + LaserMix*).
We conduct ablations on the Components Analysis by Cylinder-Mix are reliable and of high quality. This indicates that the labels augmented from labeled point clouds. Experimental results demonstrate that the former outperforms the Mean Teacher approach in LaserMix, mixes labeled and unlabeled points, with pseudos predicted by the Teacher Network. However, these pseudos are unreliable, especially in a low semi-supervised ratio. Both are fundamentally identical, except that the former has additional and reliable mixed samples derived from limited labeled data. For images, we propose the IPSL Module and scale. We compare our approach with other methods and demonstrate the effectiveness and adaptability of our model comprehensively. This third and seventh rows of Table 2 reveal that Cylinder-Mix consistently outperforms LaserMix.

Reliable potential samples. As shown in Table 2, the pretrain-retrain approach is employed in LaserMix, where only labeled point clouds are mixed and these samples are reliable. In contrast, the Mean Teacher approach, employed in LaserMix*, mixes labeled and unlabeled points, with pseudos predicted by the Teacher Network. However, these pseudos are unreliable, especially in a low semi-supervised ratio. Both are fundamentally identical, except that the former has additional and reliable mixed samples derived from labeled point clouds. Experimental results demonstrate that the former outperforms the Mean Teacher approach comprehensively. This indicates that the labels augmented by Cylinder-Mix are reliable and of high quality.

Components Analysis We conduct ablations on the Instance Position-scale Learning (IPSL) Module and Cylinder-Mix (CM) with their default settings.

Ablation on nuScenes. As shown in Table 3, we present the PQ values with semi-supervised ratios of 10%, 20% and 40%, where ✓ indicates whether IPSL and Cylinder-Mix are used. The results indicate that both modules are effective on the nuScenes dataset. IPSL performed best at a 20% proportion, bringing a 2.1% increase in PQ, while Cylinder-Mix shows a 2.7% increase at the 10% setting. Cylinder-Mix tends to have a greater impact with a smaller amount of labeled data. Moreover, we can draw the conclusion that both Cylinder-Mix and IPSL module exhibit performance gain on nuScenes dataset. Ablation on SemanticKITTI is present in the Appendix.

Table 2: Comparison of Cylinder-Mix between LaserMix.

<table>
<thead>
<tr>
<th></th>
<th>1%</th>
<th>10%</th>
<th>20%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>baseline</td>
<td>46.5</td>
<td>57.0</td>
<td>59.3</td>
<td>60.4</td>
</tr>
<tr>
<td>baseline+LaserMix</td>
<td>46.0</td>
<td>56.5</td>
<td>59.0</td>
<td>59.7</td>
</tr>
<tr>
<td>baseline+LaserMix*</td>
<td>47.0</td>
<td>58.4</td>
<td>60.0</td>
<td>59.9</td>
</tr>
<tr>
<td>baseline+Cylinder-Mix</td>
<td>49.0</td>
<td>59.9</td>
<td>60.8</td>
<td>61.2</td>
</tr>
<tr>
<td>baseline+Cylinder-Mix</td>
<td>51.2</td>
<td>61.8</td>
<td>62.6</td>
<td>64.7</td>
</tr>
<tr>
<td>baseline+LaserMix</td>
<td>49.7</td>
<td>62.3</td>
<td>63.9</td>
<td>64.8</td>
</tr>
<tr>
<td>baseline+LaserMix*</td>
<td>50.4</td>
<td>64.1</td>
<td>64.7</td>
<td>65.5</td>
</tr>
<tr>
<td>baseline+Cylinder-Mix</td>
<td>52.8</td>
<td>64.8</td>
<td>64.9</td>
<td>65.9</td>
</tr>
</tbody>
</table>

Table 3: Ablation of IPSL and Cylinder-Mix on nuScenes.

<table>
<thead>
<tr>
<th>baseline</th>
<th>IPSL</th>
<th>CM</th>
<th>10%</th>
<th>20%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>67.8</td>
<td>71.9</td>
<td>75.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>68.4</td>
<td>74.0</td>
<td>76.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>68.5</td>
<td>74.6</td>
<td>76.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>70.9</td>
<td>75.6</td>
<td>77.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Ablation study of IPSL on nuScenes. SS-detector and fully detector are trained with semi-supervised and fully nuScenes, while KITTI-detector from SemanticKITTI.

Instance Position-Scale Learning Module We conduct ablation experiments for the IPSL to demonstrate its strong generality, where even when using detectors from different datasets or detectors with slightly lower accuracy on partial datasets, the performance is comparable.

Underdemanding Detector. First, IPSL does not require high accuracy from the detector. As shown in Table 4, comparing IPSL(SS-detector) and IPSL(fully detector), the mAP of the detector trained with a partial dataset is slightly lower than that of the fully trained detector by 31.2%. However, their PQ results are comparable.

From Different datasets. Second, IPSL still works with detectors from different datasets. Due to the class-agnostic nature of the heatmap in IPSL, the detector trained on SemanticKITTI can be used to predict boxes for nuScenes images. The similarity between IPSL(KITTI-detector) and IPSL(fully detector) in Table 4 further confirms the universality of the detector of IPSL.

From LLM masks. Third, comparing IPSL (GSA mask) with the baseline and IPSL (fully detector), we observe that utilizing masks predicted by a large model, Ground-Segment-Anything (GSA), remains effective in IPSL. This further validates the generality of the IPSL module and aligns with our initial motivation for designing IPSL.

Fine-grained Analysis of Module Parameters We perform ablations on the parameters of the IPSL module and Cylinder-Mix, including $P_{center}$ for IPSL and $p_{cylmix}$ and region size for Cylinder-Mix, to ensure their rationality. Detailed results and analyses can be found in the Appendix.

Conclusion

In this paper, we exploit the latent labels from the original labels for semi-supervised multi-modal point cloud panoptic segmentation. For LiDAR data, we introduce a novel data augmentation to generate more and reliable point clouds from limited labeled data. For images, we propose the IPSL module to learn and fuse the information of instance position and scale. We compare our approach with other methods and demonstrate the effectiveness and adaptability of our model by applying a large-scale architecture to our proposed module. Our approach provides inspiration for understanding and mining deeper information from LiDAR-Camera data.
Acknowledgments
This work is supported by the National Natural Science Foundation of China (No.62222602, 62302167, U23A20343, 62106075, 62176224,62176092), Development Project of Ministry of Industry and Information Technology (ZYZB-23-990-016), Shanghai Sailing Program under Grant (23YF1410500), Natural Science Foundation of Shanghai (23ZR1420400), Natural Science Foundation of Chongqing (CSTB2023NSCQ-JQX0007, CSTB2023NSCQ-MSX0137), CCF-Tencent Rhino-Bird Young Faculty Open Research Fund (RAGR20230121), CAAI-Huawei MindSpore Open Fund.

References


