# LiDAR-Camera Panoptic Segmentation via Geometry-Consistent and Semantic-Aware Alignment

Zhiwei Zhang[1†] Zhizhong Zhang[2,3†] Qian Yu[2] Ran Yi[1] Yuan Xie[2,3*] Lizhuang Ma[1,2*]
[1]School of Electronic Information and Electrical Engineering, Shanghai Jiaotong Univeristy, China
[2]Department of Computer Science and Engineering, East China Normal University, China
[3]Chongqing Institute of East China Normal University, China
{zhangzw12319, ranyi, lzma}@sjtu.edu.cn, {qianyu, zzzhang, yxie}@cs.ecnu.edu.cn

## Abstract

*3D panoptic segmentation is a challenging perception task that requires both semantic segmentation and instance segmentation. In this task, we notice that images could provide rich texture, color, and discriminative information, which can complement LiDAR data for evident performance improvement, but their fusion remains a challenging problem. To this end, we propose LCPS, the first LiDAR-Camera Panoptic Segmentation network. In our approach, we conduct LiDAR-Camera fusion in three stages: 1) an Asynchronous Compensation Pixel Alignment (ACPA) module that calibrates the coordinate misalignment caused by asynchronous problems between sensors; 2) a Semantic-Aware Region Alignment (SARA) module that extends the one-to-one point-pixel mapping to one-to-many semantic relations; 3) a Point-to-Voxel feature Propagation (PVP) module that integrates both geometric and semantic fusion information for the entire point cloud. Our fusion strategy improves about* 6.9% *PQ performance over the LiDAR-only baseline on NuScenes dataset. Extensive quantitative and qualitative experiments further demonstrate the effectiveness of our novel framework. The code will be released at https://github.com/zhangzw12319/lcps.git.*

## 1. Introduction

3D scene perception has become an increasingly important task for a wide range of applications, including self-driving and robotic navigation. Lying in the heart of 3D vision, 3D panoptic segmentation is a comprehensive perception task composed of semantic and instance segmentation [15]. This is still challenging since it not only requires predicting semantic labels of each point for *Stuff* classes, such as *tree*, *road*, but also needs recognizing instances for *Thing* classes, e.g., *car*, *bicycle*, and *pedestrian* simultaneously.

*Corresponding authors; † equal contributions

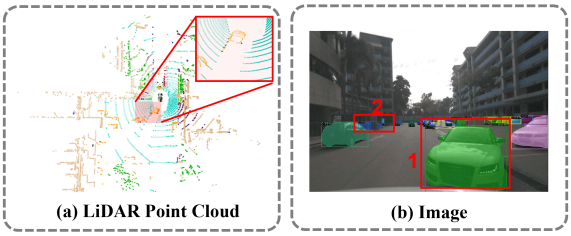


Figure 1. The distinctions between LiDAR point cloud and images. (a) The red box displays a vehicle segment (orange points) in the point cloud, where points are sparsely and unevenly distributed. (b) The lower-right green mask demonstrates a vehicle with dense texture and color features, effectively detected via [40]. The upper-left blue mask (partly occluded) shows image features that help detect small objects in the distance. Better zoomed in.

Currently, the leading 3D panoptic methods use LiDAR-only data as input sources. However, We have observed that using only LiDAR data for perception has some insufficiencies: 1) LiDAR point cloud is usually sparse and unevenly distributed, as illustrated in Figure 1 (a), making it challenging for 3D networks to capture the notable difference between the foreground and the background; 2) distant objects that occupy just a few points appear to be small in the view and cannot be effectively detected. On the contrary, images provide rich texture and color information, as shown in Figure 1 (b). This observation motivates us to use images as an additional input source to complement LiDAR sensors for scene perception. Moreover, most autonomous driving systems come equipped with RGB cameras, which makes LiDAR-Camera fusion studies more feasible.

Although LiDAR sensors and cameras complement each other, their fusion strategy remains challenging. Existing fusion strategy could be generally split into proposal-level fusion [16], result-level fusion [27], and point-level fusion [33, 12, 34], as summarized in PointAugmenting [35]. Yet, proposal-level and result-level fusion focus on integrating

2D and 3D proposals (or bounding box results) for object detection, which limits their generalizability in dense predictions like segmentation tasks. The previous point-fusion methods also suffer: 1) the asynchronous working frequency between LiDAR and camera sensors is not considered, which may result in misaligned feature correspondence; 2) point-fusion is a one-to-one fusion mechanism, and large image areas are unable to be mapped to sparse LiDAR points, resulting in the waste of abundant information from dense pixel features; *e.g.*, for a 32-beams LiDAR sensor, only about $5\%$ pixels can be mapped to correlated points, while the $95\%$ of pixel features would be dropped [23]. 3) previous point-level fusion methods [33, 12, 34] often use simple concatenation, which excludes points whose projections fall outside the image plane, as image features cannot support them.

Motivated by these insufficiencies, we propose the first LiDAR-Camera Panoptic Segmentation (LCPS) network to exploit the complementary information from multiple sensors. In this work, we propose a novel three-stage fusion strategy involving the Asynchronous Compensation Pixel Alignment (ACPA) module, Semantic-Aware Region Alignment (SARA) module, and Point-to-Voxel feature Propagation (PVP) module. The ACPA module employs ego-motion compensation operations to achieve spatial-temporal alignment between the LiDAR and camera modalities, overcoming asynchronous issues in point fusion. Then, our novel SARA module extends the one-to-one point-pixel mapping to one-to-many semantic relations, highly improving the image utilization rate. Specifically, SARA introduces Class Activation Maps (CAM) for image branch to localize semantic-related image regions for each point. Next, the PVP module replaces simple concatenation with local attention to propagate information from point-aligned pixels and regions to the entire point cloud. Points outside camera frustums can also be preserved and attached to image features. Finally, we design a Foreground Object selection Gate (FOG) module to enforce the network to learn a class-agnostic foreground object mask in addition to the semantic prediction head. This gate effectively reduces incorrect predictions and stabilizes the training process. To sum up, our main contributions are:

- To the best of our knowledge, this is the first LiDAR-Camera fusion network for 3D panoptic segmentation, which effectively exploits the complementary information of the LiDAR and image data.
- We have improved the former point-fusion approach with our novel Asynchronous Compensation Pixel Alignment (ACPA), Semantic-Aware Region Alignment (SARA), and Point-to-Voxel feature Propagation (PVP) modules. These contribute to the geometry-consistent and semantic-aware alignment between LiDAR and Camera sensors.
- We present the Foreground Object selection Gate (FOG) to reduce the incorrect predictions of confusing points, further boosting panoptic segmentation quality.
- Extensive quantitative and qualitative experiments demonstrate the effectiveness of our approach. Our fusion approach improves performance at $6.9\%$ PQ on NuScenes and $3.3\%$ PQ in SemanticKITTI compared to the LiDAR-only baseline.

## 2. Related Work

Panoptic segmentation is initially proposed from 2D vision [15], for the purpose of integrating semantic and instance segmentation. Later, research of panoptic segmentation extends to videos and LiDAR point cloud. Early work LPSAD [25] handles LiDAR panoptic segmentation via projecting points into range view and then using 2D convolution network to extract features. Although pure 2D network can boost efficiency, it also suffers performance degradation when mapping 2D predictions back to the point cloud. Later, 3D LiDAR networks are designed for this task. Generally, 3D panoptic segmentation can be divided into two categories, *i.e.,* proposal-based and proposal-free methods.

**Proposal-based 3D Panoptic Segmentation.** Proposal-based methods Panoptic-Deeplab [6] and EfficientLPS [30] predict bounding-box proposals and then merge them with semantic results to obtain panoptic predictions, following classical object detection framework[5, 9]. However, proposal-based methods tend to result in inconsistent segmentation between instance and semantic branches. Moreover, the segmentation result is susceptible to the quality of object detection.

**Proposal-free 3D Panoptic Segmentation.** Proposal-free methods abandon object proposals and predict object center and point offset instead. The post-processing module will cluster points into instance groups according to object center and point offset. DS-Net [11] proposes a dynamic-shifting mechanism of instance points toward its possible centers for Mean Shift clustering. SMAC-Seg [17] and SCAN [38] attempt to use attention module on multi-directional or multi-scale feature maps. GP-S3Net [28] constructs a dynamic graph composed of foreground clusters as graph nodes, processed by graph convolutional network for instance segmentation branch. Panoptic-Polarnet [41] projects 3D features into BEV and utilizes learnable BEV heatmap with non-maximum suppression(NMS) to predict centers. Following Panoptic-Polarnet's BEV design, Panoptic-PHNet [19] improves center and offset generation by replacing NMS with a center grouping module

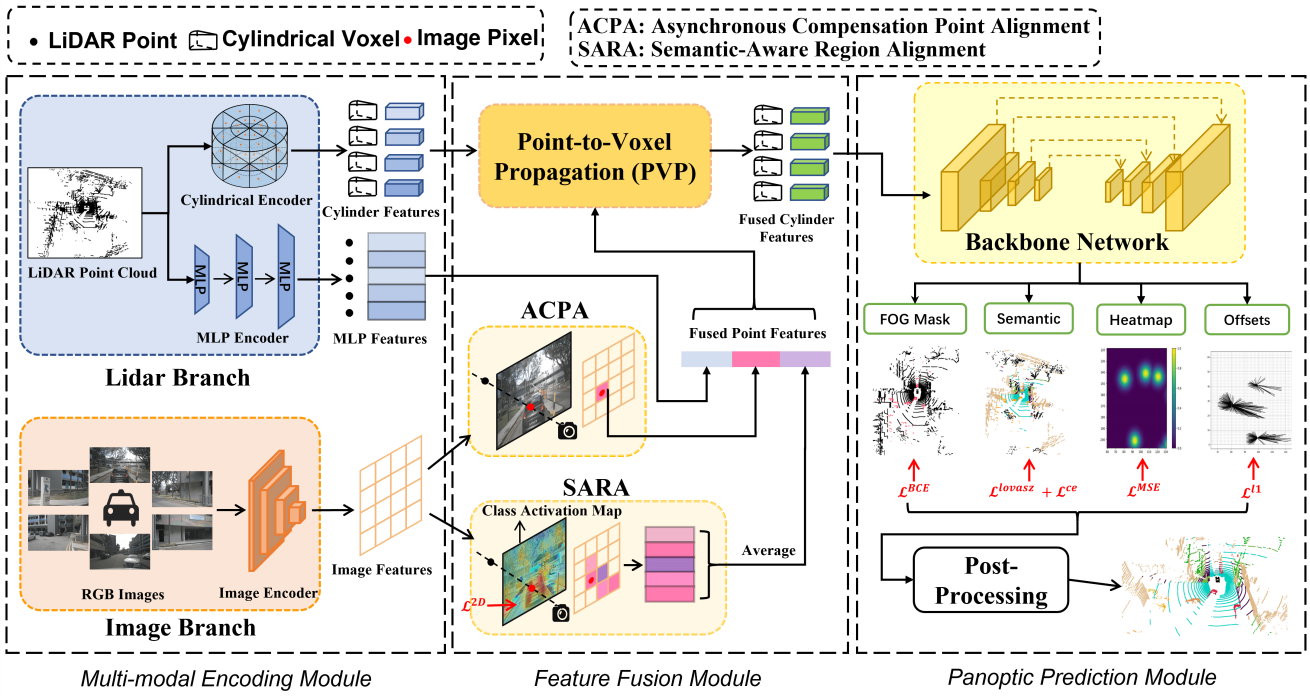


Figure 2. The overall pipeline of our LiDAR-Camera Panoptic Segmentation network (LCPS). LCPS consists of multi-modal encoding, feature fusion, and panoptic prediction modules. The encoding module extracts cylinder features, MLP features, and image features. In the fusion stage, MLP features are geometrically and semantically aligned with pixel features via ACPA and SARA. Next, the PVP module merges fused point features with original cylinder features to obtain fused ones. Finally, the panoptic prediction module yields predictions of four heads, which are post-processed to obtain panoptic segmentation results.

to merge duplicated centers, as well as augmenting offset via KNN-Transformer. For now, Panoptic-PHNet has achieved 1st place on NuScenes and SOTA performance on SemanticKITTI benchmarks.

Nevertheless, sparse and uneven LiDAR points will impose large variance for center and offset predictions in Bird-Eye-View and thus becomes a bottleneck for current SOTA approaches. RGB images can compensate for LiDAR features, which motivates us to design LCPS.

**LiDAR-Camera Fusion Models.** In object detection and semantic segmentation, pioneering research already considers modal fusion between images and LiDAR points. For example, PMF [43] attempts to project LiDAR points to the perspective view and proposes a two-branch 2D network to extract semantic features with an attentive fusion module. TransFuser [26] and TransFusion [1] consider utilizing transformers to fuse 3D LiDAR points and 2D images. DeepFusion [20] focuses on how to avoid feature misalignment when extensive data augmentation is performed in both LiDAR and camera branches. However, multi-modal panoptic segmentation has yet to be explored, accompanied by asynchronous and utilization issues.

# 3. Methodology

## 3.1. Overview

**Problem Formulation.** This paper considers 3D panoptic segmentation [7]. Formally, we denote a set of LiDAR points as $\{(x_i^{\text{3D}}, f_i^{\text{3D}})|(x_i^{\text{3D}} \in \mathbb{R}^3, f_i^{\text{3D}} \in \mathbb{R}^C\}_{i=1}^N$, where $N$, $x_i^{\text{3D}}$ and $f_i^{\text{3D}}$ represent the total number of points, 3D positions, and point features of $C$ dimensions, respectively. This task requires predicting a unique semantic class $\{\hat{y}_i^{\text{3D}}\}_{i=1}^N$ for each point and accurately identifying groups of points as foreground objects with an instance ID, denoted as $\{\text{ID}_i^{\text{3D}}\}_{i=1}^N$.

Besides, we assume that $K$ surrounding cameras, which are cheap and common, capture images associated with the LiDAR frame for LiDAR-Camera fusion. Similarly, we represent each image as a set of pixels $\{(x_{k,i}^{\text{2D}}, f_{k,i}^{\text{2D}})|(x_{k,i}^{\text{2D}} \in \mathbb{R}^2, f_{k,i}^{\text{2D}} \in \mathbb{R}^C\}_{i=1,k=1}^{i=N',k=K}$, where $N'$, $x_i^{\text{2D}}$, $f_i^{\text{2D}}$ and $k$ represent the total number of pixels, 2D positions, pixel features, and the camera index, respectively. Our primary objective in this paper is to improve panoptic segmentation performance by fully exploring the complementary information in the LiDAR and Camera sensors.

**Pipeline Architecture.** The framework in Figure 2 consists of a multi-modal encoding module, a LiDAR-Camera fea-

ture fusion module, and a panoptic prediction module. In the encoding stage, the LiDAR points are respectively encoded by a cylindrical voxel encoder and an MLP encoder, while the images are encoded using SwiftNet [36]. In the fusion stage, the MLP feature and image features, which are not strictly correlated, are first aligned through the proposed Asynchronous Compensation and Semantic-Aware Region Alignment, and then are concatenated into fused point features. Subsequently, our Point-to-Voxel Propagation module (PVP) accepts the fused point features and outputs the final cylinder representation. In the prediction stage, the backbone network includes a proposed FOG head, a semantic segmentation head, a heatmap head, and an offsets head. The latter two heads follow Panoptic-Polarnet [41], where we regress a binary object center mask and a 2D offset among bird-eye-view grids. During inference, the post-processing shifts the predicted foreground BEV grids to their nearest centers and clusters the points within the grids into instances.

### 3.2. Asynchronous Compensation Pixel Alignment

A straightforward solution [21, 33, 44] for fusing LiDAR and Camera is to establish point-to-pixel mappings, such that points can be directly projected to image planes and decorated with pixel features. However, this mapping would lead to false mapping due to the asynchronous frequency between cameras and LiDAR sensors. For instance, on NuScenes dataset, each camera operates at a frequency of 12Hz, while the LiDAR sensor operates at 20Hz.

Motivated by this, we improve the point-level fusion by incorporating additional asynchronous compensation to achieve a consistent geometric alignment over time. The fundamental idea is to transform the LiDAR points into a new 3D coordination system when the corresponding images are captured at that time. The transformation matrix is obtained by considering the ego vehicle's motion matrix. Specifically, let $t_1$ and $t_2$ denote the time when the LiDAR point cloud and the related images are captured. Then we have:

**Step-1.** Transform LiDAR points from world coordinates to ego-vehicle coordinates at time $t_1$. By multiplying the coordinate transformation matrix $\mathbf{T}_{t_1}^{\mathrm{W}\to\mathrm{V}}$ provided by the dataset, we can obtain the 3D position in the ego-vehicle coordinate system, denoted as $\hat{x}_i^{\mathrm{3D}}$.

**Step-2.** Transform LiDAR points in ego-vehicle coordinates from time $t_1$ to time $t_2$. To achieve this, a time-variant transformation matrix is required, denoted $\mathbf{T}_{t_1\to t_2}^{\mathrm{V}\to\mathrm{V}}$. However, such a matrix is often not directly available in datasets. Instead, the ego vehicle's motion matrices from the current frame to the first frame are often provided for each sliced sequence. Therefore, we can divide $\mathbf{T}_{t_1\to t_2}^{\mathrm{V}\to\mathrm{V}}$ as the product of $(\mathbf{T}_{t_2\to t_0}^{\mathrm{V}\to\mathrm{V}})^{-1}$ and $\mathbf{T}_{t_1\to t_0}^{\mathrm{V}\to\mathrm{V}}$, where $t_0$ is the time of the first frame. Using this ego-motion transformation matrix,

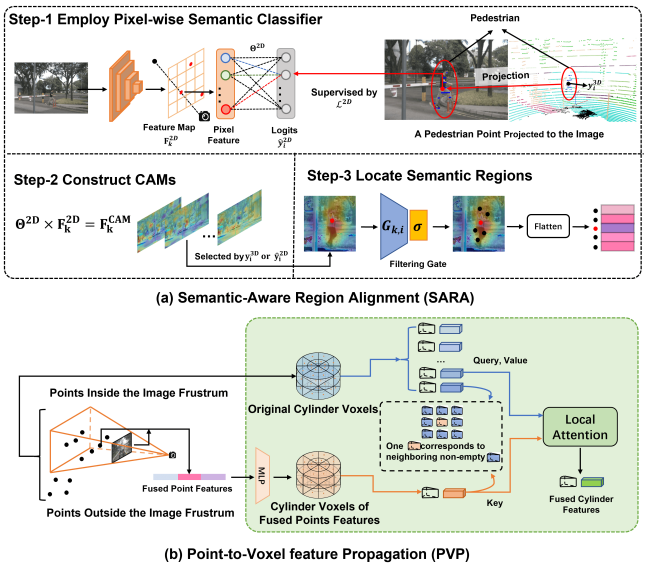


Figure 3. (a) Overview of the SARA module, which employs pixel-wise semantic classifier, constructs CAMs and locates semantic regions. (b) Overview of the PVP Module, which involves a cylindrical partition of fused point features and attentive propagation. Better zoomed in.

we obtain the point position in ego-vehicle coordinates at time $t_2$, denoted as $\tilde{x}_i^{\mathrm{3D}}$.

**Step-3.** Obtain pixel features at time $t_2$. By using camera extrinsic and intrinsic matrices ($\mathbf{E}_k$ and $\mathbf{I}_k$), we get the projected 2D position $\tilde{x}_{k,i}^{\mathrm{2D}}$ of each point in the $k_{\mathrm{th}}$ image plane at time $t_2$. After excluding the points whose projections are outside the image plane, the resulting pixel features $\{\tilde{f}_{k,i}^{\mathrm{2D}}\}_{i=1}^{N_k}$ are indexed by $\tilde{x}_{k,i}^{\mathrm{2D}}$. $N_k$ is the number of points inside the image plane ($N_k < N$).

These homogeneous transformation steps can be summarized in the following equation:

$$\begin{bmatrix} \tilde{x}_{k,i}^{\mathrm{2D}} \\ 1 \end{bmatrix} = \mathbf{I}_k \mathbf{E}_k \mathbf{T}_{t_1\to t_2}^{\mathrm{V}\to\mathrm{V}} \mathbf{T}_{t_1}^{\mathrm{W}\to\mathrm{V}} \begin{bmatrix} x_i^{\mathrm{3D}} \\ 1 \end{bmatrix}. \quad (1)$$

In summary, we obtain pixel-aligned features for each point using Equation 1. Our approach adopts ego-motion compensation via Step 2, resulting in a simple but more accurate geometric-consistent feature alignment.

### 3.3. Semantic-Aware Region Alignment

Due to the sparse nature and limited eyeshot of LiDAR point clouds, only a small fraction of image features can be matched with LiDAR points. To address this issue, we propose to find semantic-relevant regions, extending the one-to-one mapping to one-to-many relations. Inspired by *Class Activation Maps* (CAMs) [39, 24], we present a Semantic-Aware Region Alignment module by using image CAMs to localize relevant semantic regions, as illustrated in Figure 3

(a).

**Step-1.** We first introduce a pixel-wise semantic classifier $\phi^{\text{2D}}(\cdot)$ to learn the semantic information in the image branch, and define $\mathbf{\Theta}^{\text{2D}} \in \mathbb{R}^{M\times C}$ as the classifier parameters, where $M$ is the number of semantic categories. Based on the observation that projected pixels share the same semantic category with matched points, we use point labels to train the image classifier with cross-entropy loss:

$$\mathcal{L}_{\text{2D}} = -\frac{1}{N_k}\sum_{i=1}^{N_k} y_i^{\text{3D}} \log(\hat{y}_i^{\text{2D}}), \tag{2}$$

where $\hat{y}_i^{\text{2D}}$ and $y_i^{\text{3D}}$ denote the predicted pixel label and related ground-truth point label (such alignment is obtained in Section 3.2), and $N_K$ represents the number of points which can be projected into the $k$-th image plane.

**Step-2.** We use this classifier to generate the class activation maps (CAMs). Let $\mathbf{F}_k^{\text{2D}} \in \mathbb{R}^{C\times H^{\text{2D}}\times W^{\text{2D}}}$ be the image feature map extracted by the last convolution layer, and $H^{\text{2D}}$ and $W^{\text{2D}}$ are the height and width of image feature maps. We can then obtain CAMs using the following formula:

$$\mathbf{F}_k^{\text{CAM}} = \mathbf{\Theta}^{\text{2D}} \times \mathbf{F}_k^{\text{2D}}, \tag{3}$$

where $\times$ denotes the matrix multiplication. The generated CAMs are represented by $\mathbf{F}_k^{\text{CAM}} \in \mathbb{R}^{M\times H^{\text{2D}}\times W^{\text{2D}}}$. Each channel in CAM is a $H^{\text{2D}} \times W^{\text{2D}}$ heatmap related to a specific semantic category.

**Step-3.** For each LiDAR point, we use the generated CAMs to localize sets of pixels as semantic-related image regions. We design a filtering gate $\mathbf{G}_{k,i}^{y} \in \mathbb{R}^{H^{\text{2D}}\times W^{\text{2D}}}$, constructed by selecting a single heatmap of class $y$ from CAMs $F_k^{\text{CAM}}$ according to the ground-truth or predicted pixel label. The gate is controlled by subtracting a predefined confidence threshold $\tau$. Pixels with heatmap values lower than that threshold will be set to zero in $\mathbf{G}_{k,i}^{y}$. Finally, we get a set of related pixels:

$$\{f^{\text{CAM}}\}_{k,i} = \textit{Flatten}(\sigma(\mathbf{G}_{k,i}^{y} \otimes \mathbf{F}_k^{\text{2D}})), \tag{4}$$

where $\otimes$ denotes element-wise multiplication, and $\sigma$ denotes the activation function. *Flatten* function is adopted to transform features from matrix format $C \times H^{\text{2D}} \times W^{\text{2D}}$ into a set format $(H^{\text{2D}}W^{\text{2D}}) \times C$, followed by discarding zero vectors which is filtered by $G_{k,i}$. Consequently, we obtain a set of pixel features $\{f^{\text{CAM}} \in \mathbb{R}^{C}\}_{k,i}$ for each LiDAR point $i$ and each camera $k$.

We finally average the set of region features to a single vector, then concatenate it with the MLP output and pixel-aligned features to constitute the fused point features. In summary, unlike one-to-one pixel alignment via pure geometric projection, the image regions are directly collected in a one-to-many semantic-aware manner.

### 3.4. Point-to-Voxel Feature Propagation

Image features seem to not support the points outside the camera frustum; therefore, these points are usually excluded [29, 33, 12]. To overcome this problem, we propose the Point-to-Voxel Feature Propagation to integrate both geometric and semantic information for the entire point cloud. To this end, we choose cylindrical voxels as the bridge to complete the fusion process since the tensor shape of the voxel representation is invariant to the alteration of point numbers, which naturally provides an alignment between the original point cloud and the image-related point cloud subset.

As shown in Figure 3 (b), a cylindrical encoder first encodes the original point cloud into voxels. Meanwhile, for the fused point features, we first align their channel dimensions with the original voxel using MLP, and then divide these fused points into another set of cylindrical voxels, where features will be scattered and pooled within the same voxel to obtain voxel features. A noticeable observation is that a LiDAR point may have alignment with more than one camera, resulting in multiple fused point features of a single point. Therefore, we treat such multiple features as multiple points at the same 3D position during voxelization. Then we propagate the voxels of the fused point features (denoted as $\vartheta^{\text{im}}$) to the original cylindrical voxels (denoted as $\vartheta^{\text{p}}$) using modified local attention [32]. In this attention mechanism, each voxel $\vartheta^{\text{p}}$ acts as queries $Q$, while the neighboring 27 $\vartheta^{\text{im}}$ voxels act as keys $K$ and values $V$. Then the computation is given by:

$$\text{Att}(\vartheta^{\text{p}}, \vartheta^{\text{im}}, \vartheta^{\text{im}}) = \text{Softmax}(\frac{QK^T}{\sqrt{C}})V, \tag{5}$$

where $C$ is the channel dimensions. After that, we add the attentive voxels with original $\vartheta^{\text{p}}$ to make a residual connection, as shown in the following equation:

$$\vartheta = \text{Att}(\vartheta^{\text{p}}, \vartheta^{\text{im}}, \vartheta^{\text{im}}) + \vartheta^{\text{p}}. \tag{6}$$

Through this attentive propagation, information from the entire point cloud and multiple cameras is comprehensively integrated into a single cylindrical voxel representation $\vartheta$.

### 3.5. Improved Panoptic Segmentation

Here we briefly describe the Foreground Object selection Gate (FOG) head and loss functions for panoptic prediction. Other implementation details are displayed in Section 4.2 and the Appendix.

**Foreground Object Selection Gate.** In Panoptic-PolarNet [41], the panoptic network diverges into three prediction heads for semantic labels, centers, and offset prediction. However, we find that semantic predictions largely affect the final quality of panoptic segmentation. This is

because the center and offset head only provide class-agnostic predictions, while accurate semantic information is required for post-processing to cluster foreground grids into the nearest object centers. Inspired by [22], we propose FOG, a Foreground Object Selection Gate, to enhance the original semantic classifier. FOG is a binary classifier aiming to differentiate foreground objects. Given voxel features obtained from the backbone network as $\vartheta^{\text{b}} \in \mathbb{R}^{H\times W\times Z}$, FOG predicts a class-agnostic binary mask $y^{\text{FOG}} \in [0,1]^{H\times W\times Z}$, which is supervised by binary cross-entropy loss $\mathcal{L}^{\text{BCE}}$. As a result, the foreground mask complements the semantic head by filtering out background points in the post-processing period.

**Loss Designs.** The total loss is derived in the following equation:

$$\mathcal{L}^{\text{total}} = \alpha_1(\mathcal{L}^{\text{CE}}+\mathcal{L}^{\text{Lovasz}})+\alpha_2\mathcal{L}^{\text{MSE}}+\alpha_3\mathcal{L}^{\text{L1}}+\alpha_4\mathcal{L}^{\text{BCE}}+\alpha_5\mathcal{L}^{\text{2D}}. \tag{7}$$

The top four terms are based on Panoptic-Polarnet [41]. $\mathcal{L}^{\text{CE}}$ and $\mathcal{L}^{\text{Lovasz}}$ represent cross-entropy loss and Lovasz loss [4] for semantic supervision. $\mathcal{L}^{\text{MSE}}$ is a Mean-Squared-Error (MSE) loss for BEV center heatmap regression. $\mathcal{L}^{\text{L1}}$ is an L1 loss for BEV offset regression. In addition, the last two terms are new in this paper. $\mathcal{L}^{\text{BCE}}$ represents a binary entropy loss used for FOG head, and $\mathcal{L}^{\text{2D}}$ is a pointly-supervised loss for region-fusion, given by Equation 2. $\alpha_2$ and $\alpha_3$ are set to 100 and 10 respectively, while the other three weights are set to 1.

# 4. Experiments

In this section, we evaluate our proposed LiDAR-Camera Panoptic Segmentation network on NuScenes [7] and SemanticKITTI [3] dataset, making comparisons with recent state-of-the-art methods.

## 4.1. Datasets and Evaluation Metric

**NuScenes** is a large-scale multi-modal dataset for autonomous driving. It contains a 32-beam LiDAR, 5 Radars, 6 RGB cameras and maps, covering 1000 real-world driving scenes of 4 locations in Boston and Singapore. There are 850 annotated scenes for training and 150 for testing. The panoptic annotations contain 10 *Thing* classes, 6 *Stuff* classes and 1 class for noisy labels.

**SemanticKITTI** is a pioneering outdoor dataset presenting the panoptic segmentation tasks on LiDAR data [3, 2, 8]. It provides a 64-beam LiDAR sensor and two front-view cameras. The dataset contains 8 *Thing* classes and 11 *Stuff* classes, consisting of 19130 frames for training, 4071 for validation, and 20351 for testing.

**Evaluation Metrics.** We assess the panoptic segmentation via panoptic quality (PQ), segmentation quality (SQ), and recognition quality (RQ) [7]. Metrics with superscripts th and st (*e.g.,* $PQ^{th}$) represent *Thing* or *Stuff* classes performance respectively. Meanwhile, we also provide semantic segmentation metrics (mIoU) [3].

## 4.2. Implementation Details

**Backbone Network.** Cylinder3D [42, 11] is adopted as our backbone network in Figure 2 due to its reliable LiDAR perception ability for cylinder voxel representation. As for NuScenes, the entire point cloud is divided into $480\times360\times32$ voxels for $[-100\text{m}\sim100\text{m}, 0\sim2\pi, -5\sim3\text{m}]$ polarized volume of the scenery. As for SemanticKITTI, we only change the perception distance from 100m to 60m.

**Settings and Hyper-parameters.** Following common practice [19, 38], we apply random flip augmentation along the $y$-axis for the point cloud and images accordingly, and random rotation for the point cloud only. These LiDAR augmentations are adopted after precomputing the point-pixel alignment. The performance gains from data augmentations are already included in LiDAR-only baseline results for fair comparisons, as shown in the first line of Table 4. We train our model for 120 epochs with a batch size of 2, using Adam optimizer [14]. The initial learning rate is 0.004 and will be reduced to 0.0004 after 100 epochs. For SARA described in Section 3.3, the filtering parameter $\tau$ is set to 0.7. During inference, all operations are performed in BEV grids, where centers are picked from a dynamic heatmap using non-maximum-suppression with a kernel size of 5 and a value threshold of 0.1. Other setting details are described in the Appendix.

## 4.3. Main Results

In this section, we make extensive comparisons with other state-of-the-art methods and our LiDAR-only baseline. Specifically, the baseline network excludes the image branch, feature fusion module, and FOG in Figure 2.

**Results on NuScenes.** In Table 1, our approach outperforms the best Panoptic-PHNet [19] by a margin of 5.1% PQ (79.8% vs. 74.7%) in validation set. Primarily, we achieve a large gain of 4.3% RQ and 7.1% $\text{RQ}^{\text{th}}$, which mainly increases the overall accuracy. Compared with the LiDAR-only baseline, our methods show a significant improvement of 6.9% PQ in total, demonstrating the effectiveness of our LiDAR-Camera fusion strategy. As for the test set, we also achieve comparable SOTA results with Panoptic-PHNet [19] without using test-time augmentation and 6.7% PQ increase compared with our LiDAR-only baseline.

Evidence from the class-wise comparison on NuScenes validation set also consolidates the effectiveness of our fusion strategy. Figure 4 shows that an overall improvement among various *Thing* and *Stuff* categories can be witnessed. Specifically, for *Thing* objects like *bicycle*, *bus*, *construction vehicle*, *motorcycle*, and *traffic cone*, our method out-

| Method | PQ | PQ† | SQ | RQ | $PQ^{th}$ | $SQ^{th}$ | $RQ^{th}$ | $PQ^{st}$ | $SQ^{st}$ | $RQ^{st}$ | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DS-Net [11] | 42.5 | 51.0 | 83.6 | 50.3 | 32.5 | 83.1 | 38.3 | 59.2 | 84.4 | 70.3 | 70.7 |
| GP-S3Net [28] | 48.7 | 60.3 | 61.3 | 63.7 | 61.6 | 86.4 | 71.7 | 43.8 | 51.8 | 60.8 | 61.8 |
| PanopticTrackNet [13] | 50.0 | 57.3 | 80.9 | 60.6 | 45.1 | 80.3 | 52.4 | 58.3 | 81.9 | 74.3 | 63.1 |
| EfficientLPS [30] | 62.0 | 65.6 | 83.4 | 73.9 | 56.8 | 83.2 | 68.0 | 70.6 | 83.8 | 83.6 | 65.6 |
| SCAN [38] | 65.1 | 68.9 | 85.7 | 75.3 | 60.6 | 85.7 | 70.2 | 72.5 | 85.7 | 83.8 | 77.4 |
| Panoptic-PolarNet [41] | 67.7 | 71.0 | 86.0 | 78.1 | 65.2 | 87.2 | 74.0 | 71.9 | 83.9 | 84.9 | 69.3 |
| SMAC-Seg HiRes [17] | 68.4 | 73.4 | 85.2 | 79.7 | 68.0 | 87.3 | 77.2 | 68.8 | 83.0 | 82.1 | 71.2 |
| CPSeg HR [18] | 71.1 | 75.6 | 85.5 | 82.5 | 71.5 | 87.3 | 81.3 | 70.6 | 83.6 | 83.7 | 73.2 |
| Panoptic PH-Net [19] | 74.7 | 77.7 | 88.2 | 84.2 | 74.0 | 89.0 | 82.5 | **75.9** | **86.8** | **86.9** | 79.7 |
| PUPS [31] | 74.7 | 77.3 | 89.4 | 83.3 | 75.4 | 91.8 | 81.9 | 73.6 | 85.3 | 85.6 | - |
| **LCPS (Baseline)** | 72.9 | 77.6 | 88.4 | 82.0 | 72.8 | 90.1 | 80.5 | 73.0 | 85.5 | 84.5 | 75.1 |
| **LCPS (Full)** | **79.8** | **84.0** | **89.8** | **88.5** | **82.3** | **91.7** | **89.6** | 75.6 | 86.7 | 86.5 | **80.5** |

Table 1. 3D panoptic segmentation results on NuScenes validation set. The evaluation metric is provided in PQ%.

| Method | PQ | PQ† | SQ | RQ | $PQ^{th}$ | $SQ^{th}$ | $RQ^{th}$ | $PQ^{st}$ | $SQ^{st}$ | $RQ^{st}$ | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EfficientLPS [30] | 62.4 | 66.0 | 83.7 | 74.1 | 57.2 | 83.6 | 68.2 | 71.1 | 83.8 | 84.0 | 66.7 |
| Panoptic-PolarNet [41] | 63.6 | 67.1 | 84.3 | 75.1 | 59.0 | 84.3 | 69.8 | 71.3 | 84.2 | 83.9 | 67.0 |
| Panoptic PH-Net [19] | **80.1** | **82.8** | **91.1** | **87.6** | **82.1** | 93.0 | 88.1 | **76.6** | **87.9** | **86.6** | **80.2** |
| **LCPS (Baseline)** | 72.8 | 76.3 | 88.6 | 81.7 | 72.4 | 90.2 | 80.0 | 73.5 | 86.1 | 84.6 | 74.8 |
| **LCPS (Full)** | 79.5 | 82.3 | 90.3 | **87.7** | 81.7 | 92.2 | **88.6** | 75.9 | 87.3 | 86.3 | 78.9 |

Table 2. 3D panoptic segmentation results on NuScenes test set. Our result is compared with other methods without test-time augmentation and ensemble operations.

performs Panoptic-PHNet by a large margin (9.3% on average for 5 *Thing* classes), which demonstrates the ability of our approach to distinguish the sparse, distant and rare objects by taking advantages from image features.

**Results on SemanticKITTI.** Here, we list the comparison results of the SemanticKITTI validation set in Table 3. Since SemanticKITTI has only two cameras in the front view, fewer points can be matched with image features compared with NuScenes, thus increasing the difficulty of LiDAR-Camera fusion. Nevertheless, we discover an increase of 3.3% PQ over the LiDAR-only baseline, demonstrating the robustness and effectiveness of our fusion strategy.

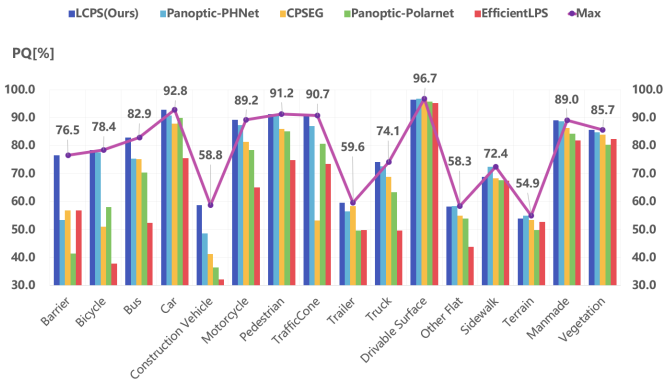


Figure 4. Class-wise PQ% results on NuScenes validation set.

| Method | PQ | PQ† | SQ | RQ | mIoU |
|---|---|---|---|---|---|
| PanopticTrackNet [13] | 40.0 | - | 73.0 | 48.3 | 53.8 |
| DS-Net [11] | 57.7 | 63.4 | 77.6 | 68.0 | 63.5 |
| Panoptic-PolarNet [41] | 59.1 | 64.1 | 78.3 | 70.2 | 64.5 |
| EfficientLPS [30] | 59.2 | 65.1 | 75.0 | 69.8 | 64.9 |
| Panoptic PH-Net [19] | 61.7 | - | - | - | 65.7 |
| GP-S3Net [28] | 63.3 | 71.5 | 81.4 | 75.9 | 73.0 |
| PUPS [31] | **64.4** | **68.6** | **81.5** | **74.1** | - |
| **LCPS (Baseline)** | 55.7 | 65.2 | 74.0 | 65.8 | 61.1 |
| **LCPS (Full)** | 59.0 | 68.8 | 79.8 | 68.9 | 63.2 |

Table 3. 3D panoptic segmentation results on SemanticKITTI validation set.

| ACPA | SARA | PVP | SC | FOG | PQ |
|---|---|---|---|---|---|
| | | | | | 72.9 |
| ✓ | | | ✓ | | 76.8 |
| ✓ | | ✓ | | | 77.5 |
| ✓ | ✓ | ✓ | | | 79.2 |
| ✓ | ✓ | ✓ | | ✓ | 79.8 |

Table 4. Ablation study on NuScenes validation set. The SC represents Simple Concatenation compared to PVP.

## 4.4. Ablation Study

To analyze the source of remarkable performance improvements, we conduct an ablation study on various components of our approach on NuScenes validation set. As depicted in Table 4, we divide the ablation study into the following three parts.

**Ablation on Fusion Modules.** We separately verify the effectiveness of the Asynchronous Compensation Point Alignment (ACPA), Semantic-Aware Region Alignment

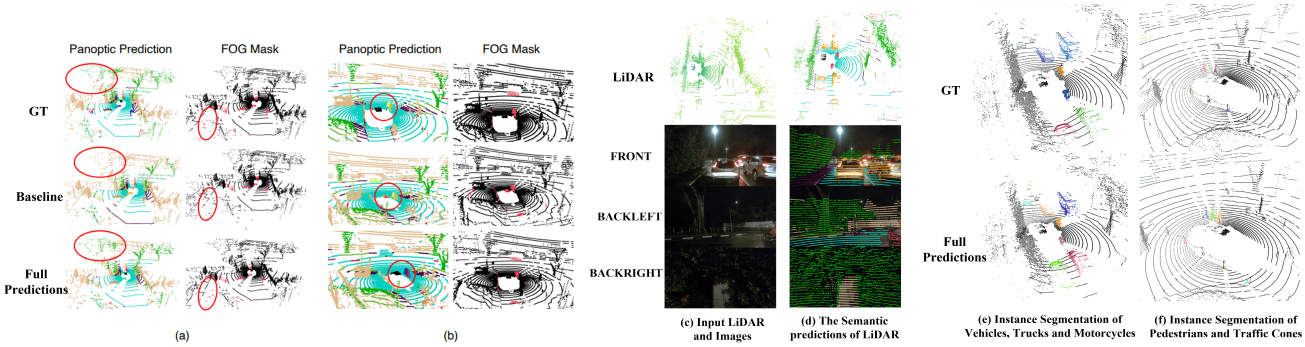


Figure 5. The overview of qualitative results from NuScenes validation set. (a) and (b) are visualization comparisons among the ground-truth (denoted as GT), the baseline predictions (Baseline), and full LCPS predictions (Full). Red circles emphasize the notable differences. We find that various *Thing* and *Stuff* objects can be predicted more accurately. (c) and (d) demonstrate semantic segmentation quality at nighttime. (e) and (f) verify the robust instance segmentation ability of our network. Better zoomed in.

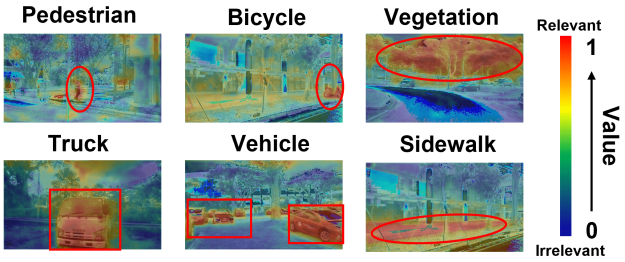


Figure 6. The visualization results of semantic-aware regions filtered by CAMs.

(SARA), and Point-to-Voxel feature Propagation (PVP). It is observed that the ACPA with simple concatenation (SC) could bring an improvement of $3.9\%$ PQ (contrasting line 1 and line 2) and an improvement of $4.6\%$ PQ with PVP module (contrasting line 1 and line 3). Another $1.7\%$ PQ gain can be achieved combined with SARA (contrasting line 3 and line 4). It verifies our designs on geometry-consistent and semantic-aware LiDAR-Camera fusion strategy.

**Ablation on FOG Mask.** We test the influence of the FOG mask and observe the improvement of $0.6\%$ PQ (contrasting line 4 and line 5). It suggests that FOG Mask may bring additional supervision to the backbone network and further augments the semantic prediction in post-processing grouping.

### 4.5. Qualitative results and the Discussion

**Visualization of Panoptic Predictions.** In Figure 5, we evaluate our visual predictions compared among ground-truth (GT), baseline, and full network (Full Predictions). The following observation can be made: 1) Our architecture achieves effective semantic and instance segmentation among challenging scenarios, like crowds of pedestrians and vehicles (see Figure 5 (a)(b)(e)(f)); 2) Our LiDAR-Camera Fusion strategies can achieve robust segmentation quality at nighttime with the complementary information from surrounding cameras (see Figure 5 (c)(d)); 3) FOG can help filter confusing points and noise points, making segmentation quality more robust (see Figure 5 (a)(b)).

**Visualization of Class Activation Maps.** We further verify the quality of generated Class Activation Maps (CAMs) in Figure 6, which constitute the semantic-aware regions in images. The red color illustrates higher semantic correlations, while the blue color refers to lower ones. It demonstrates that our SARA module generates highly correlated alignment among various categories, effectively extending the one-to-one mapping to semantic-aware one-to-many relations.

## 5. The Conclusion

In this paper, we are the first to propose the geometry consistent and semantic-aware LiDAR-Camera Panoptic Network. As a new paradigm, we effectively exploit complementary information from LiDAR-Camera sensors and make essential efforts to overcome asynchronous and utilization problems via Asynchronous Compensation Point Alignment (ACPA), Semantic-Aware Region Alignment (SARA), Point-to-Voxel feature Propagation (PVP), and Foreground Object selection Gate (FOG) mask. These modules enhance the overall discriminability and performance. We hope that our thought-invoking multi-modal fusion practice can benefit future research.

**Acknowledgement** The research is supported by National Natural Science Foundation of China (No.62222602, No.61972157 and No.72192821), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Science and Technology Commission (21511101200), Shanghai Sailing Program (22YF1420300 and 23YF1410500), Natural Science Foundation of Chongqing (No.CSTB2023NSCQ-JQX0007), CCF-Tencent Open Research Fund (RAGR20220121), Young Elite Scientists Sponsorship Program by CAST

(2022QNRC001) and CAAI-Huawei MindSpore Open Fund.

## References


[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022. 3

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss. Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. *The International Journal on Robotics Research*, 40(8-9):959–967, 2021. 6

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 6

[4] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4413–4421, 2018. 6

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2

[6] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 2

[7] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. 3, 6

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 6

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 11

[11] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 13090–13099. Computer Vision Foundation / IEEE, 2021. 2, 6, 7, 11

[12] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. EPNet: Enhancing point features with image semantics for 3d object detection. In *Computer Vision – ECCV 2020*, pages 35–52. Springer International Publishing, 2020. 1, 2, 5

[13] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Scalability in Autonomous Driving*, 2020. 7

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[15] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 1, 2

[16] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018. 1

[17] Enxu Li, Ryan Razani, Yixuan Xu, and Bingbing Liu. SMAC-seg: LiDAR panoptic segmentation via sparse multi-directional attention clustering. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, may 2022. 2, 7

[18] Enxu Li, Ryan Razani, Yixuan Xu, and Bingbing Liu. CPSeg: Cluster-free panoptic segmentation of 3d LiDAR point clouds. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2023. 7

[19] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11809–11818, 2022. 2, 6, 7, 11, 12

[20] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17182–17191, 2022. 3

[21] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Computer Vision – ECCV 2018*, pages 663–678. Springer International Publishing, 2018. 4

[22] Long Ang Lim and Hacer Yalim Keles. Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications*, 23(3):1369–1380, 2020. 6

[23] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view

representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781. IEEE, 2023. 2

[24] R Austin McEver and BS Manjunath. Pcams: Weakly supervised semantic segmentation using point supervision. *arXiv preprint arXiv:2007.05615*, 2020. 4

[25] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. Lidar panoptic segmentation for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8505–8512. IEEE, 2020. 2

[26] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021. 3

[27] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum PointNets for 3d object detection from RGB-d data. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018. 1

[28] Ryan Razani, Ran Cheng, Enxu Li, Ehsan Taghavi, Yuan Ren, and Liu Bingbing. GP-S3Net: Graph-based panoptic sparse semantic segmentation network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16076–16085, 2021. 2, 7

[29] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. MVX-net: Multimodal VoxelNet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, may 2019. 5

[30] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, 2021. 2, 7

[31] Shihao Su, Jianyun Xu, Huanyu Wang, Zhenwei Miao, Xin Zhan, Dayang Hao, and Xi Li. PUPS: Point cloud unified panoptic segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):2339–2347, jun 2023. 7

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[33] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. PointPainting: Sequential fusion for 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 1, 2, 4, 5

[34] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. PointAugmenting: Cross-modal augmentation for 3d object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021. 1, 2

[35] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11794–11803, 2021. 1, 12

[36] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, and Song Bai. Swiftnet: Real-time video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1296–1305, June 2021. 4

[37] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, and Song Bai. Swiftnet: Real-time video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1296–1305, 2021. 11

[38] Shuangjie Xu, Rui Wan, Maosheng Ye, Xiaoyi Zou, and Tongyi Cao. Sparse cross-scale attention network for efficient LiDAR panoptic segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):2920–2928, jun 2022. 2, 6, 7

[39] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. 4

[40] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. In *arXiv preprint arXiv:2103.07461*, 2021. 1

[41] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021. 2, 4, 5, 6, 7, 11, 12

[42] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. *arXiv preprint arXiv:2011.10033*, 2020. 6, 11

[43] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16280–16290, 2021. 3

[44] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d LiDAR semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2021. 4

# Appendix

## A. Further Implementation Details

In this section, we further elaborate on the implementation details of our LCPS. Section 3.1 in the main paper explains that the LiDAR branch consists of point and voxel streams. We employ three MLP layers for the point stream to extract point-level features with an output dimension of 64, 128, and 256 channels. Following processing by ACPA and SARA, the fused point features are compressed to 16 channels to match the voxel features. As for the voxel stream, the cylindrical encoder maps original point features (XYZ-axis, remission, reflections, etc) to 16 dimensions. After PVP module, features from the voxel stream and point stream are merged into cylinders and then fed into Cylinder3D [42] backbone network. In Cylinder3D, four layers of down-sampling 3D convolutions with BatchNorm and ReLU activation are applied to the fused voxel features, transforming the channel numbers of voxel features to 32, 64, 128, and 256, respectively. Finally, the voxel feature dimension is compressed back to 128 after pooling layers and remains at this dimension in the subsequent four up-sampling layers of Cylinder3D.

Regarding the image branch, images from multiple cameras are concatenated and scaled to 0.4 of the original size. The SwiftNet-18[37] network comprises four pairs of down-sampling and up-sampling layers. The four down-sampling layers transform features maps to $64 \times 320 \times 180$, $128 \times 160 \times 90$, $256 \times 80 \times 45$, and $512 \times 40 \times 23$, respectively. Then, a multi-scale spatial pooling module [10] is utilized to compress the four feature maps to 128 channels. Eventually, the other 4 up-sampling layers symmetrically up-sample the feature map to the input size for following geometric-consistent and semantic-aware alignment and fusion.

## B. Further Discussion

**Visual Ablations on Asynchronous Compensation.** Here we further provide qualitative comparisons of asynchronous compensation in Figure 7. The first and the second lines are visual results without or with asynchronous compensation respectively. The leftmost three columns demonstrate the effectiveness, especially for foreground objects of various sizes and distinct geometric shapes. For instance, the most apparent improvement is exhibited in the second leftmost column, where few points can be mapped to distant and marginal trucks, greatly enhancing the robustness of LiDAR-Camera fusion. We also demonstrate a typical failed case here to illustrate the limitation. When the ego-vehicle slows down, or objects come at the front or back view, it is possible that the asynchronous compensation almost makes no difference because the time gap or the changes of view angles is small.

**Discussions on Time and Memory Cost.** We compare the time and memory cost in Table 5 with other SOTA approaches if their projects are open-source or if such information is provided in their papers. Our LCPS full model is slightly slower than LiDAR-only methods (including our LiDAR-only baseline). Interestingly, adding image branches does not essentially drop the FPS since we choose lightweight ResNet-18 as the image backbone. The parameter number of our LiDAR-only baseline is high since we replace the backbone BEV U-Net in Panoptic-Polarnet [41] with Cylinder3D [42]. Similarly, the parameter number of DS-Net [11] is also above 50M since it adopts Cylinder3D as the backbone network too.

| | FPS(Hz) | Params(M) |
|---|---|---|
| DS-Net [11] | 3.2 | 56.5 |
| Panoptic-PolarNet [41] | 11.6 | 13.8 |
| Panoptic-PHNet [19] | 11.0 | - |
| LCPS(LiDAR-Only) | 8.6 | 65.9 |
| LCPS(Full) | 8.3 | 77.7 |

Table 5. Results of FPS and parameter scales on SemanticKITTI.

**More In-Depth Analysis on Metrics.** We provide further in-depth analysis of our experimental results. It appears that some *Stuff* metrics, e.g., $PQ^{st}$ and mIoU, are slightly lower than Panoptic-PHNet (0.3%-0.7%) on NuScenes val. and test set. However, compared to the LiDAR-only baseline, our approach boosts the performance in terms of $PQ^{st}$, $SQ^{st}$, $RQ^{st}$, and mIoU consistently. This phenomenon reflects that the improvement on *Stuff* is not as noticeable as *Thing* objects and further indicates that the LiDAR-Image fusion has more benefits on *Thing* objects. The possible reason is that *Thing* objects often have fewer points than *Stuff*; thus, images may provide more crucial information for the former.

Our experimental results do not obviously surpass SOTA methods since our baseline is relatively weak. Our baseline project is built on the current highest open-source benchmark, Panoptic-Polarnet [41], while other SOTA methods have not released their codes yet. We reproduce the Panoptic-Polarnet and get 67.7% PQ on NuScenes, and 55.7% PQ on SemanticKITTI. Then we further improve the NuScenes baseline to 72.9% using data augmentations as stated in Section 4.2. Based on this baseline, our fusion strategy obtains +6.9%, +6.7%, and +3.3% PQ improvement on NuScenes validation, NuScenes test, and SemanticKITTI validation set, reported in the main content. After submission, we improve our baseline on SemanticKITTI by using rare *Stuff* copy-paste augmentation, demonstrating higher overall performances, as shown in Table 7. We provide the improved version here for additional reference and analysis. The fusion improvement on SemanticKITTI is lower than NuScenes since SemanticKITTI has only two front-view cameras; thus, the number of matched points is lower than NuScenes, as illustrated in

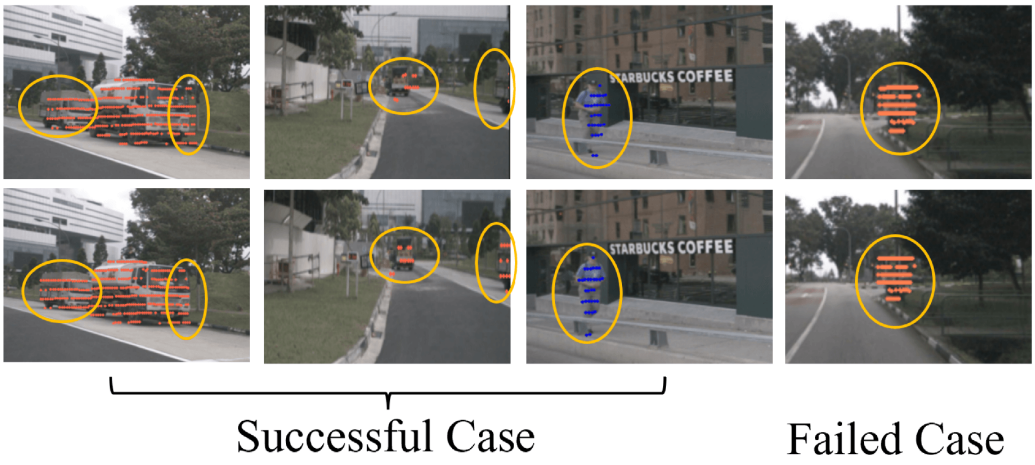


Figure 7. Visual comparisons of asynchronous compensation. The first and the second lines are visualizations without or with asynchronous compensation respectively. The leftmost three columns demonstrate the effectiveness, especially for foreground objects of various sizes and geometric shapes. The last column specifies that the asynchronous compensation almost makes no difference when the time gap is small or at the front view.

Table 6. Additionally, the improvement on a higher SemanticKITTI baseline is lower than the original one because heavy data augmentation (we add extra rare stuff augmentation in order to achieve a higher baseline) may diminish the benefits of LiDAR-Camera fusion, which is also reported in previous research on detection tasks like PointAugmenting [35].

As for mIoU, it is mainly evaluated for semantic segmentation, and we include it following previous research [19, 41]. Better PQ simultaneously needs better semantic segmentation ability (mIoU) and instance segmentation quality. Therefore, a model of high mIoU may perform worse in terms of PQ. Our method can achieve comparable mIoU performance with SOTA Panoptic-PHNet in NuScenes, while worse in SemanticKitti due to the weak baseline issue. Besides, our fusion strategy consistently improves PQ and mIoU in both NuScenes and SemanticKitti datasets compared to the LiDAR-only baseline.

| | NuScenes | SemanticKITTI |
|---|---|---|
| Matched Points | 17182 | 39780 |
| Total Points | 34720 | 120387 |
| Percentage | 52.2 % | 33.2 % |

Table 6. Statistics on the averaged number of points matched to images per frame.

| Methods | Improved (Val.) | | Improved (Test.) | |
|---|---|---|---|---|
| | PQ | mIoU | PQ | mIoU |
| LCPS(LiDAR-Only) | 60.6 | 66.8 | 57.8 | 62.0 |
| LCPS(Full) | 61.4 | 67.5 | 58.8 | 62.8 |

Table 7. Results of the fusion methods on the improved baseline of SemanticKITTI.

**Ablation Study on Perception Distance.** As our backbone network adopts a cylindrical voxel representation, we need to set the perception distance of the scene volume, which is defined as the radial distance from the LiDAR sensor to objects or points. Setting the perception distance too close or too far is sub-optimal for training because a close distance setting may miss some small objects far away and diminish PQ performance, while a far distance setting may involve more noise points and disturb training stability.

In our experimentation on the NuScenes validation set (as shown in Table 8), we find that as the perception distance increases, the performance initially improves and then declines. The result shows that $\pm 100$ meters and $\pm 120$ meters yield the highest PQ scores, while $\pm 80$ meters produce the best mIoU. Intuitively, $\pm 80$ meters can be the valid distance at which the LiDAR sensor is able to accurately detect objects in NuScenes, while approximately $\pm 150$ meters is the farthest perception distance. Based on these findings, we ultimately choose $\pm 100$ meters as the moderate perception distance for NuScenes.

**Correction Ability.** When we review the visualization results, one interesting observation is that our model appears to correct segmentation errors in the ground-truth labels. Due to the utilization of bounding boxes and semantic labels in rule-based scripts for automatically generating panoptic labels, errors in ground-truth labels are commonly observed. Hence, it is essential for our network to boost more generalizability and avoid overfitting with ground-truth labels during the training process.

As illustrated in Figure 9, our LCPS network is capable of correcting ground-truth errors by preserving the intact shape of an instance. Figure 9 (a) demonstrates that the predicted truck segmentation retains the top edge area since it is spatially and geometrically proximate to the truck segments below. Similarly, in a sequence of frames in Fig-

| Distance (±, [meters]) | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120 | 130 | NFV |
|---|---|---|---|---|---|---|---|---|---|---|
| PQ [%] | 70.6 | 72.2 | 72.5 | 72.8 | 72.8 | **72.9** | 72.2 | **72.9** | 72.3 | 70.5 |
| mIoU [%] | 74.3 | 74.6 | 74.0 | **75.2** | 74.5 | 75.1 | 74.4 | 74.5 | 74.4 | 73.7 |

Table 8. The ablation of perception distance on NuScenes validation set. The experiment is tuned on our LiDAR-only baseline network. NFV represents No Fixed Volume, which means we select the farthest point (usually $\geq$ 170m) as the distance for each LiDAR scan rather than a fixed distance.

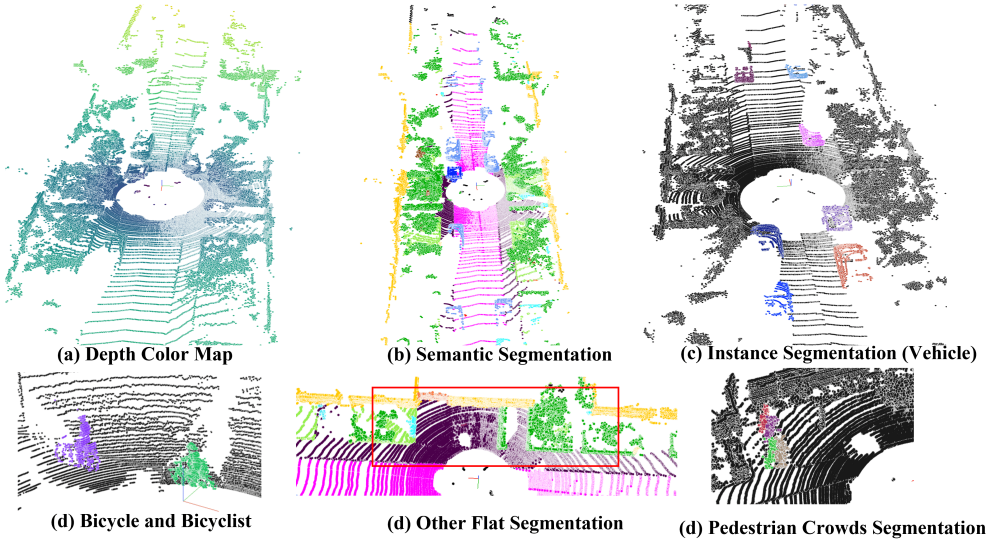


Figure 8. Visualization results of SemanticKITTI validation set.

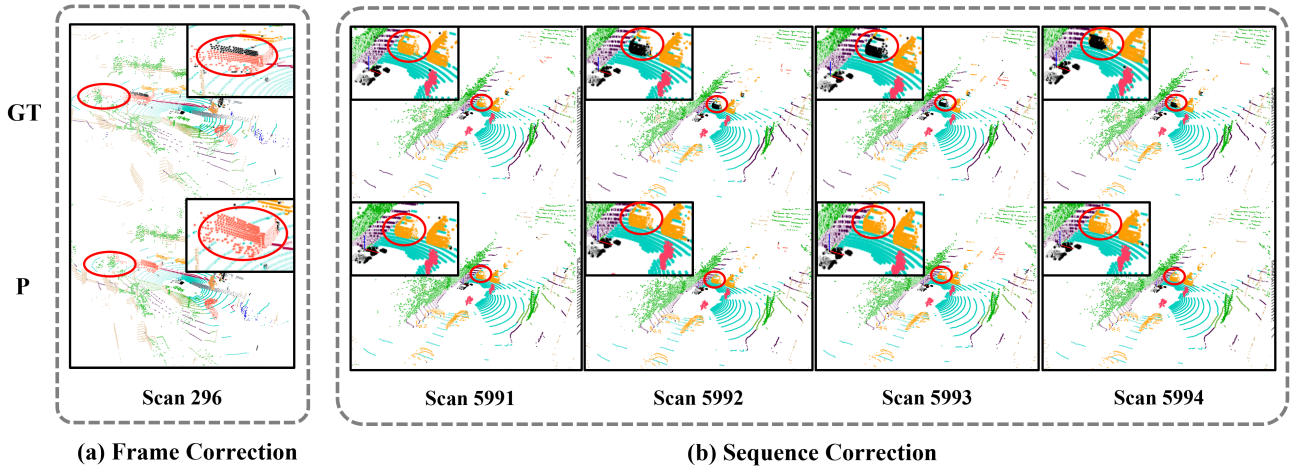


Figure 9. The visualization of correction ability between ground-truth (GT, first line) and our model predictions (P, second line). The results are from the NuScenes validation set, where the scan number below represents the sample index and red circles highlight notable differences. (a) shows that our model can segment an intact segmentation of truck, even though the ground-truth labels overlook the top area. (b) demonstrates that our model can consistently segment an entire vehicle during a sequence of frames over time, while the ground-truth labels miss the back surface when the ego-car advances at a high speed.

ure 9 (b), the rule-based ground-truth label generation results in the absence of the back surface of the vehicle, as the

ego-car advances at high speed. Nevertheless, our network still maintains consistency in predicted segmentation over time, which serves as compelling evidence that our network obtains robust feature representation of objects by leveraging LiDAR and image features, enabling it to correct false ground-truth labels.

**Further Qualitative Results.** We provide further visualization results on the validation set of NuScenes (Figure 10 and Figure 11 ) and SemanticKITTI (Figure 8) dataset to detailedly demonstrate the panoptic segmentation ability of our network. In Figure 10, we display the objects whose perspective projections are within the single image and compare ground-truth labels and predictions in 3D and perspective view. Our network effectively recognizes small objects (such as *bicycle* and *motorcycle*) and rare objects (such as *trailers* and *construction vehicle*). Especially in the right-construction vehicle sub-figure, our segmentation quality is slightly better than ground-truth labels at the position of robotic arms. Regarding Figure 11, we compare the visualization results of objects across multiple images. We primarily select challenging scenarios such as crowding (pedestrian and car) and severe occlusion (truck). For example, the truck segmentation is largely occluded by walls, which severely damages the geometric structure in LiDAR scenes and feature completeness in images. Under such conditions, our network can correctly segment most of the truck instances while missing one truck only (which is occluded by the orange construction vehicles). Moreover, for pedestrian segmentation, our network additionally segments two more occluded figures in the middle image column, although it wrongly recognizes two tiny figures as one person in the leftmost image column.

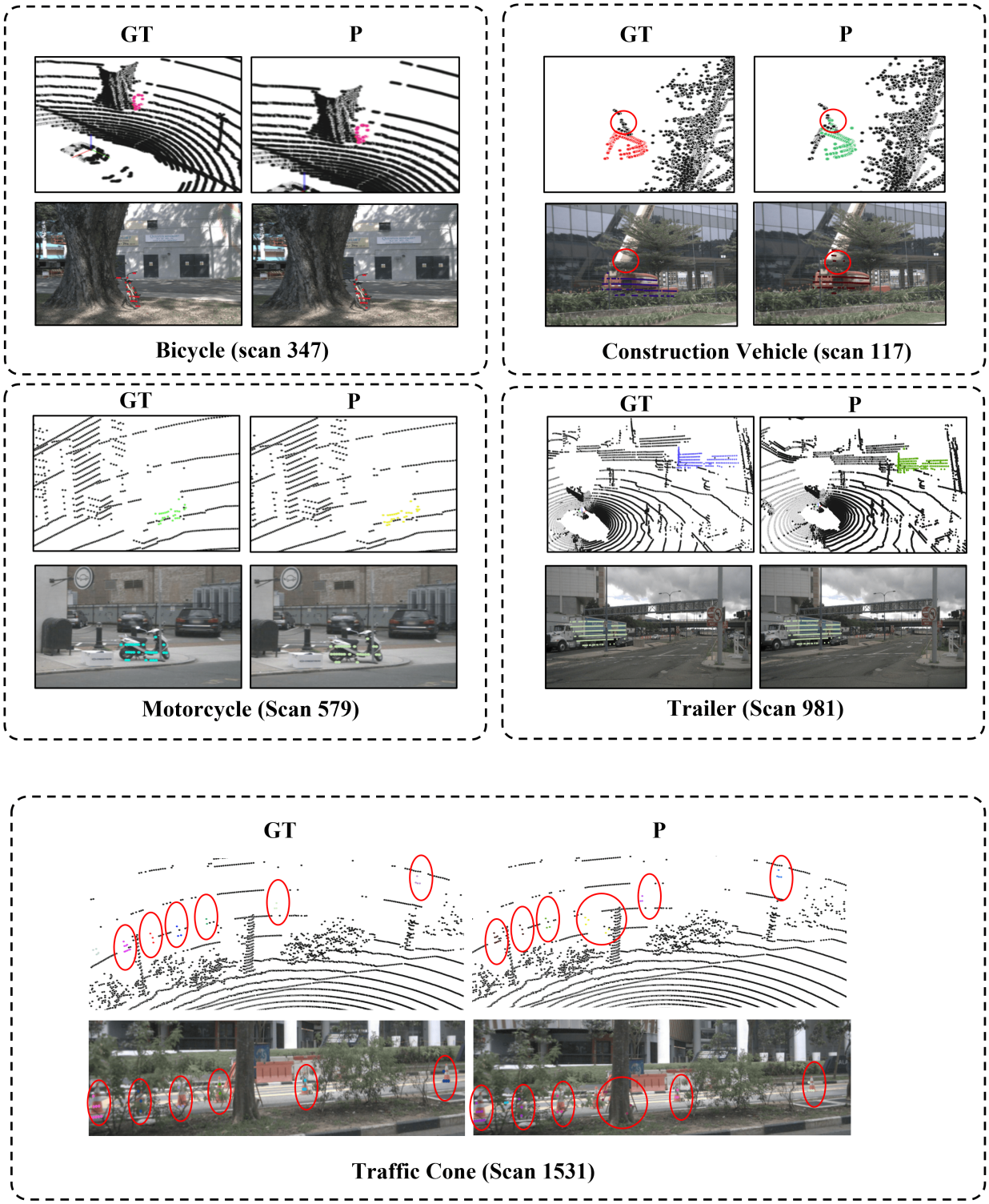


Figure 10. Visualization results of foreground objects. GT represents ground-truth labels, while P represents predictions of our LCPS. Text such as "Back" and "FRONT_LEFT" refers to the specific camera sensor. In this figure, the perspective projections of object segmentation are within the same image. Generally, our network achieves accurate segmentation results over these small and distant objects, such as *bicycle* and *motorcycle*, or rare objects like *construction vehicle*.


Figure 11. Visualization results of foreground objects. GT represents ground-truth labels, while P represents predictions of our LCPS. Texts like "Back" and "FRONT_LEFT" refer to the specific camera sensor. This figure shows that most objects of diverse types and spatial locations across images can be consistently identified.