Center Focusing Network for Real-Time LiDAR Panoptic Segmentation

Xiaoyan Li[1,2] Gang Zhang[3] Boyue Wang[1,2] Yongli Hu[1,2] Baocai Yin[1,2]
[1]Beijing Municipal Key Lab of Multimedia and Intelligent Software Technology
[2]Beijing Institute of Artificial Intelligence, Faculty of Information Technology,
Beijing University of Technology, Beijing 100124, China
[3]Mogo Auto Intelligence and Telemetics Information Technology Co. Ltd.
{xiaoyan.li,wby,huyongli,ybc}@bjut.edu.cn zhanggang11021136@gmail.com

Abstract

LiDAR panoptic segmentation facilitates an autonomous vehicle to comprehensively understand the surrounding objects and scenes and is required to run in real time. The recent proposal-free methods accelerate the algorithm, but their effectiveness and efficiency are still limited owing to the difficulty of modeling non-existent instance centers and the costly center-based clustering modules. To achieve accurate and real-time LiDAR panoptic segmentation, a novel center focusing network (CFNet) is introduced. Specifically, the center focusing feature encoding (CFFE) is proposed to explicitly understand the relationships between the original LiDAR points and virtual instance centers by shifting the LiDAR points and filling in the center points. Moreover, to leverage the redundantly detected centers, a fast center deduplication module (CDM) is proposed to select only one center for each instance. Experiments on the SemanticKITTI and nuScenes panoptic segmentation benchmarks demonstrate that our CFNet outperforms all existing methods by a large margin and is 1.6 times faster than the most efficient method.

1. Introduction

Panoptic segmentation [18] combines both semantic segmentation and instance segmentation in a single framework. It predicts semantic labels for the uncountable *stuff* classes (*e.g. road*, *sidewalk*), while it simultaneously provides semantic labels and instance IDs for the countable *things* classes (*e.g. car*, *pedestrian*). The LiDAR panoptic segmentation is one of the bases for the safety of autonomous driving, which employs the point clouds collected by the Light Detection and Ranging (LiDAR) sensors to effectively depict the surroundings. Existing LiDAR panoptic segmentation methods first conduct semantic segmentation, and then achieve instance segmentation for the *things* categories in two ways, the proposal-based and proposal-free methods.

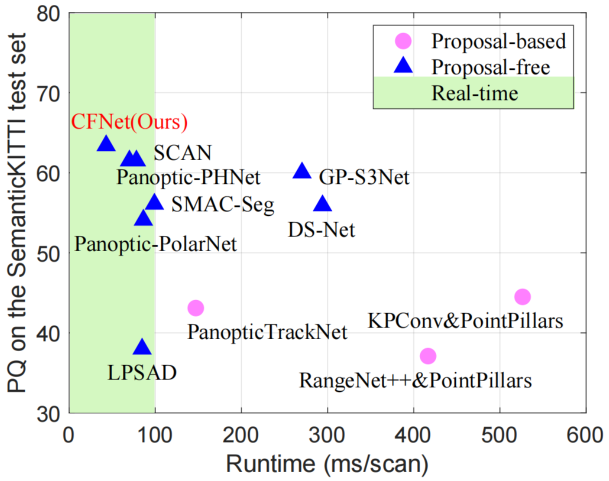
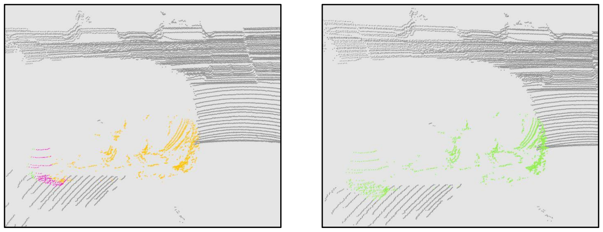


Figure 1. PQ vs. runtime on the SemanticKITTI test set. Runtime measurements are taken on a single NVIDIA RTX 3090 GPU. The panoptic quality (PQ) is introduced in section 4.1.

The proposal-based methods [17, 31, 37] adopt a two-stage process similar to the well-known Mask R-CNN [14] in the image domain. It first generates object proposals for the *things* points by using 3D detection networks [19, 30] and then refines the instance segmentation results within each proposal. As shown in Fig. 1, these methods are usually complicated and hardly achieve real-time processing, owing to their sequential multi-stage pipelines.

The proposal-free frameworks [13, 15, 21, 22, 29, 35, 39] are more compact. To associate the *things* points with instance IDs, these methods usually leverage the instance centers. Specifically, they regress the offsets from the points to their corresponding centers, and then adopt the class-agnostic center-based clustering modules [13, 15, 29] or the bird's-eye view (BEV) center heatmap [22, 35, 39]. However, two problems exist in these methods. First, for center feature extracting and center modeling, the non-existent in-



(a) CFNet without CFFE (b) CFNet with CFFE

Figure 2. Instance segmentation of a car. Without our CFFE, the car is split into parts (a), while the CFFE significantly alleviates this problem (b). Different colors represent different instances.

stance centers increase the difficulty, considering that the LiDAR points are usually surface-aggregated [35] and an instance center is imaginary in most cases. As shown in Fig. 2(a), the difficulty often results in the fault that one instance is incorrectly split into several parts. Second, for exploiting the redundantly detected centers, the clustering modules (*e.g*. MeanShift, DBSCAN) are too time-consuming to support the real-time autonomous driving perception systems, while the BEV center heatmap cannot distinguish objects with different altitudes in the same BEV grid.

For accurate and fast LiDAR panoptic segmentation, a proposal-free center focusing network (CFNet) is proposed. For better encoding center features, a novel center focusing feature encoding (CFFE) is proposed to generate center-focusing feature maps by shifting the *things* points to fill in the non-existent instance centers for more accurate predictions (as shown in Fig. 2(b)). For center modeling, the CFNet not only decomposes the panoptic segmentation task into the widely-used semantic segmentation and center offset regression, but also proposes a new confidence score prediction for indicating the accuracy of the center offset regression. Subsequently, for the detected centers exploiting, a novel center deduplication module (CDM) is designed to select one center for a single instance. The CDM keeps the predicted centers with higher confidence scores, while suppressing the ones with lower confidence. Finally, instance segmentation is achieved by assigning the shifted *things* points to the closest center. For efficiency, the proposed CFNet is built on the 2D projection-based segmentation paradigm. Our contributions are as follows:

- A proposal-free CFNet is proposed to achieve accurate and fast LiDAR panoptic segmentation by solving the bottleneck problems of center modeling and center-based clustering in previous methods.
- The CFFE is proposed to alleviate the difficulty of modeling the non-existent instance centers and the CDM is designed to efficiently keep one center for each instance.
- The proposed CFNet is evaluated on the nuScenes and SemanticKITTI LiDAR panoptic segmentation benchmarks. Our CFNet achieves the state-of-the-art performance with a real-time inference speed.

2. Related Work

The LiDAR panoptic segmentation methods usually adopt the LiDAR semantic segmentation networks as the backbone and jointly optimize the semantic segmentation and instance segmentation tasks. The results of the two tasks are fused to generate the final panoptic segmentation results. The LiDAR semantic segmentation backbone is first introduced. Then, the panoptic segmentation methods, achieving instance segmentation upon the semantic segmentation backbone, are grouped into the proposal-based and proposal-free methods, and are further discussed.

LiDAR semantic segmentation backbone. The backbone network for LiDAR semantic segmentation can be divided into three groups: point-based, voxel-based, and 2D projection-based methods. The point-based methods [16, 27, 28, 33] directly operate on the raw point clouds but are extremely time-consuming due to the expensive local neighbor searching. The voxel-based methods [7, 8, 32, 40] discretize the point clouds into structured voxels, where sparse 3D convolutions are applied. These methods are still difficult to meet the real-time applications, although they achieve the highest accuracy. The 2D projection-based methods are more efficient since most of their computation is done in the 2D space, such as range view (RV) [10,25,34], bird's-eye view (BEV) [19], polar view [38], and multi-view fusion [1, 23]. For fast LiDAR panoptic segmentation, the Panoptic-PolarNet [39], SMAC-Seg [20], and LPSAD [24] all adopt the 2D projection-based backbone. A recent real-time semantic segmentation method, namely CPGNet [23], is a 2D projection-based one that explores an end-to-end multi-view fusion framework by fusing the features from the point, BEV, and RV.

Proposal-based methods. The proposal-based methods conduct instance segmentation through a two-stage complicated process. They first detect the foreground instances and then refine the instance segmentation results independently within each detected bounding box. Based on the Mask R-CNN [14] for instance segmentation, the MOPT [17] and EfficientLPS [31] insert a semantic branch to achieve panoptic segmentation on the range view (RV). Recently, LidarMultiNet [37] unifies LiDAR-based 3D object detection, semantic segmentation, and panoptic segmentation in a single framework to reduce the computation cost by sharing a strong voxel-based backbone.

Proposal-free methods. The proposal-free methods usually apply the class-agnostic clustering to the *things* points to conduct instance segmentation. The LPSAD [24] clusters points into instances by regressing the center off-

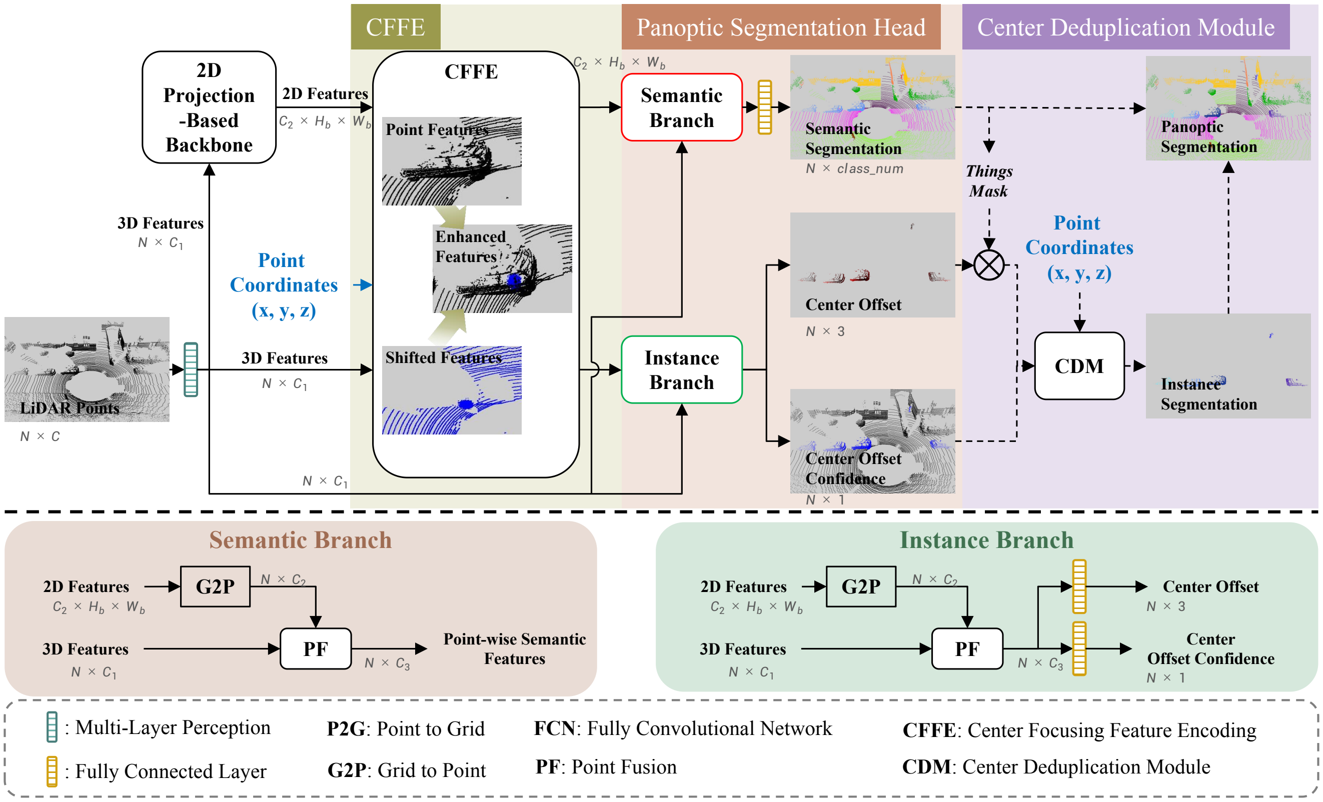


Figure 3. The overview of our CFNet. It consists of four steps: 1) the 2D projection-based backbone extracts features on the 2D space; 2) the proposed center focusing feature encoding (CFFE) mimics and enhances the non-existent instance center features; 3) the panoptic segmentation head predicts the output results; 4) the proposed center deduplication module (CDM) achieves instance segmentation that is fused to generate the final panoptic segmentation results. The dashed lines indicate that the operations are only used during inference.

sets. The Panoster [13] directly predicts the instance IDs from a learning-based clustering module, where the time-consuming DBSCAN [11] is used to refine the instance segmentation results. The DS-Net [15] proposes a learnable dynamic shifting module that adjusts the kernel functions of the MeanShift [9] to handle instances with various sizes. The GP-S3Net [29] constructs a graph convolutional network (GCN) on the over-segmentation clusters to identify instances. The SMAC-Seg [20] proposes a sparse multi-directional attention clustering and center-aware repel loss for instance segmentation. These methods adopt the time-consuming clustering methods. To further accelerate the algorithm, recently, Panoptic-PolarNet [39], SCAN [35], and Panoptic-PHNet [22] adopt the BEV center heatmap and center offsets for instance segmentation. However, the BEV center heatmap is also costly and confuses on z-axis.

3. Approach

The input of the LiDAR panoptic segmentation task is the LiDAR point clouds, given by a set of unordered points $\boldsymbol{P} = \{(\boldsymbol{p}_i, \boldsymbol{f}_i)\}_{i=0}^{N-1}$ (where $\boldsymbol{p}_i = (x, y, z)$ is the 3D coordinates in the cartesian space and $\boldsymbol{f}_i$ denotes additional LiDAR point features, *e.g. intensity*). The task aims to assign a set of labels $\boldsymbol{L} = \{(s_i, inst_i)\}_{i=0}^{N-1}$ to these points, where s_i is the semantic label (*e.g. road*, *building*, *car*, *pedestrian*), and $inst_i$ is the instance ID of the i^{th} point. Moreover, s_i can be divided into the uncountable *stuff* classes (*e.g. road*, *building*) and countable *things* classes (*e.g. car*, *pedestrian*). The instance IDs of *stuff* points are set as 0.

To predict the labels $\boldsymbol{L}$ of the input LiDAR point clouds $\boldsymbol{P}$, our CFNet decomposes this process into four steps, as shown in Fig. 3: 1) an off-the-shelf 2D projection-based backbone is applied to efficiently extract features on the 2D spaces (*e.g.* BEV, RV); 2) a novel center focusing feature encoding (CFFE) is used to generate center-focusing feature maps for more accurate predictions of the instance centers; 3) the panoptic segmentation head fuses the features from the 3D points and 2D spaces to predict the semantic segmentation results, the center offsets, and the confidence scores of the center offsets, respectively; 4) during inference, a post-process is conducted to produce the panoptic segmentation results, where a novel center deduplication module (CDM) operates on the shifted *things* points to se-

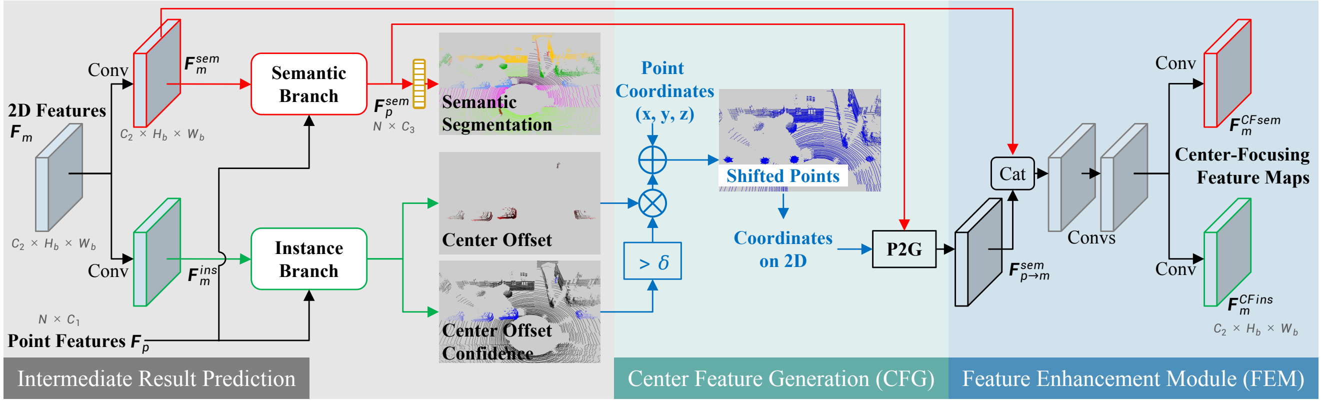


Figure 4. The proposed center focusing feature encoding (CFFE). The "Conv" represents a 2D convolution with 3 × 3 kernels, a batch normalization, and a ReLU layer. The details of the semantic branch and instance branch are shown in Fig. 3. The blue arrows are coordinate-related operations.

lect only one center for a single instance.

The design of the backbone in the first step is beyond the scope of the paper, and the last three mentioned steps are illustrated in detail in the following subsections.

3.1. Center Focusing Feature Encoding

As mentioned above, the LiDAR points of an object are usually surface-aggregated, especially for *car* and *truck* categories, resulting in the fact that the center of an object is imaginary and does not exist in the LiDAR point clouds. To encode the features of the non-existent centers, a novel center focusing feature encoding (CFFE) is proposed, which takes the 2D features from the backbone and the 3D point coordinates as inputs and generates the enhanced center-focusing feature maps as shown in Fig. 3.

The CFFE module consists of three steps, including intermediate result prediction, center feature generation, and feature enhancement module, as shown in Fig. 4.

Intermediate result prediction. In this step, the CFFE predicts intermediate results (including the semantic segmentation, center offset, and its confidence scores) according to the 2D features $\boldsymbol{F}_m$ and 3D point features $\boldsymbol{F}_p$ for subsequent center feature simulation. Specifically, two convolution layers are applied on the 2D features $\boldsymbol{F}_m$ independently, to generate semantic features $\boldsymbol{F}_m^{sem}$ and instance features $\boldsymbol{F}_m^{ins}$ (m is the specific 2D view, such as RV [10, 25, 34], BEV [19], and polar view [38].)

$$\boldsymbol{F}_m^{sem} = Conv(\boldsymbol{F}_m; \boldsymbol{\theta}_1), \tag{1}$$

$$\boldsymbol{F}_m^{ins} = Conv(\boldsymbol{F}_m; \boldsymbol{\theta}_2), \tag{2}$$

where $Conv$ denotes sequential 2D convolution, batch normalization, and ReLU operations, and $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are their learnable parameters. Then, the semantic branch generates per-point 3D semantic features $\boldsymbol{F}_p^{sem}$ by fusing the point features $\boldsymbol{F}_p$ and 2D semantic features $\boldsymbol{F}_m^{sem}$,

$$\boldsymbol{F}_p^{sem} = Seg(\boldsymbol{F}_p, \boldsymbol{F}_m^{sem}; \boldsymbol{\theta}_3), \tag{3}$$

where Seg is the semantic branch, and $\boldsymbol{\theta}_3$ is the parameters.

Finally, the intermediate semantic results $\hat{\boldsymbol{S'}} = \{\hat{s'}_i\}_{i=0}^{N-1}$ is produced based on $\boldsymbol{F}_p^{sem}$. The intermediate results of center offsets $\hat{\boldsymbol{O'}} = \{\hat{\boldsymbol{o'}}_i\}_{i=0}^{N-1}$ and confidence scores $\hat{\boldsymbol{C'}} = \{\hat{c'}_i\}_{i=0}^{N-1}$ are predicted by the instance branch with the point features $\boldsymbol{F}_p$ and 2D instance features $\boldsymbol{F}_m^{ins}$,

$$\hat{\boldsymbol{S'}} = FC(\boldsymbol{F}_p^{sem}; \boldsymbol{\theta}_4), \tag{4}$$

$$\hat{\boldsymbol{O'}}, \hat{\boldsymbol{C'}} = Ins(\boldsymbol{F}_p, \boldsymbol{F}_m^{ins}; \boldsymbol{\theta}_5), \tag{5}$$

where Ins is the instance branch, and FC represents the fully-connected layer. The structure and the training objective of semantic and instance branches are the same as those in the panoptic segmentation head and are elucidated in Fig. 3 and section 3.2.

Center feature generation (CFG). In this step, the CFFE generates the shifted center features $\boldsymbol{F}_{p\to m}^{sem}$ by shifting the 3D semantic point features $\boldsymbol{F}_p^{sem}$ to the predicted centers according to the mentioned intermediate results. First, the coordinates of a predicted center $\boldsymbol{p}_i^{shift}$ are computed by,

$$\boldsymbol{p}_i^{shift} = \boldsymbol{p}_i + \hat{\boldsymbol{o'}}_i * \mathbb{1}[\hat{c'}_i > \delta], \tag{6}$$

where $\boldsymbol{p}_i$ is the original 3D LiDAR point coordinates, and $\mathbb{1}[\hat{c'}_i > \delta]$ is a binary indicator of whether the confidence $\hat{c'}_i$ is greater than $\delta = 0.2$. In other words, it does not shift the *stuff* points or the *things* points with low confidence scores.

Then, the shifted 3D points $\{\boldsymbol{p}_i^{shift}\}_{i=0}^{N-1}$ are served as the new coordinates of the feature points, and the Point to Grid (P2G) operation in [23] re-projects the 3D semantic features $\boldsymbol{F}_p^{sem}$ onto the 2D projected feature maps $\boldsymbol{F}_{p\to m}^{sem}$ with this new coordinates,

$$\boldsymbol{F}_{p\to m}^{sem} = P2G(\boldsymbol{F}_p^{sem}; \{\boldsymbol{p}_i^{shift}\}_{i=0}^{N-1}). \tag{7}$$

Compared with $\boldsymbol{F}_p^{sem}$, the feature maps $\boldsymbol{F}_{p\to m}^{sem}$ pay more attention to the imaginary centers since most of the *things* points have been shifted to their predicted centers.

Feature enhancement module (FEM). The CFFE finally fuses the semantic feature maps $\boldsymbol{F}_m^{sem}$ and the reprojected shifted center feature maps $\boldsymbol{F}_{p\to m}^{sem}$ to generate center-focusing semantic feature maps $\boldsymbol{F}_m^{CFsem}$ and instance feature maps $\boldsymbol{F}_m^{CFins}$, which are used by the subsequent semantic and instance branches for more accurate predictions. The enhancement module is composed of a simple concatenation operation and several convolution layers, whose details are in the supplementary material.

In addition, for application of the multi-view fusion backbone (*e.g*. [23] shown in the supplementary material), the feature maps $\boldsymbol{F}_m$, $\boldsymbol{F}_m^{sem}$, $\boldsymbol{F}_m^{ins}$, $\boldsymbol{F}_{p\to m}^{sem}$ (*e.g*. $m \in \{\text{RV}, \text{BEV}\}$) of each view are computed independently according to the above process. Then, they are fused in the point fusion (PF) module [23] to generate integrated 3D point features and per-point prediction.

3.2. Panoptic Segmentation Head

For better modeling the instance center, the panoptic segmentation head uses a semantic branch to predict the semantic segmentation, and an instance branch to estimate both the center offsets and the newly introduced confidence scores, given the center-focusing semantic and instance feature maps. The architectures of both branches follow the segmentation head in [23] and are briefly introduced, while their training objectives are elaborated on.

Semantic branch. For a per-point prediction, the semantic branch first applies the Grid to Point (G2P) operations to acquire the 3D point-wise features from the 2D semantic feature maps. Then, a PF module fuses the point-wise features from the G2P operation and the original 3D points to generate point-wise semantic features. The details of the G2P and PF can be found in [23]. After obtaining the point-wise semantic features, a fully connected (FC) layer is used to predict the final per-point semantic result $\hat{\psi}_i^j$ ($\hat{\psi}_i^j \in [0, 1]$). $\hat{\psi}_i^j$ represents the probability of the i^{th} LiDAR point belonging to the j^{th} class. The predicted semantic label $\hat{s}_i$ is acquired by selecting the most probable class $\hat{s}_i = \arg\max_j \hat{\psi}_i^j$.

Referring to the CPGNet [23], the same loss functions are adopted, including weighted cross entropy (WCE) loss $\mathcal{L}_{wce}$, Lovász-Softmax loss $\mathcal{L}_{ls}$ [4], and transformation consistency loss $\mathcal{L}_{tc}$.

Instance branch. Similar to the semantic branch, the instance branch also adopts the G2P operation and a PF module to acquire point-wise instance features. An FC layer is applied to predict the per-point center offsets $\hat{\boldsymbol{o}}_i$. The ground-truth for $\hat{\boldsymbol{o}}_i$ is the offset vector $\boldsymbol{e}_i - \boldsymbol{p}_i$ from the i^{th} point $\boldsymbol{p}_i$ to its corresponding instance center $\boldsymbol{e}_i$.

For center offset regression, the loss function for optimizing $\hat{\boldsymbol{o}}_i$ only considers the *things* classes and is formulated as follows,

$$\mathcal{L}_{\boldsymbol{o}_i} = \begin{cases} ||\hat{\boldsymbol{o}}_i - (\boldsymbol{e}_i - \boldsymbol{p}_i)||_2 & \text{if } s_i \in \textit{things} \text{ classes}, \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

where $\boldsymbol{e}_i$ is the axis-aligned center of the instance [15]. Then, the losses are summarized as the following,

$$\mathcal{L}_o = \frac{1}{N^{things}} \sum_{i=0}^{N-1} \mathcal{L}_{\boldsymbol{o}_i}, \tag{9}$$

where N and N^{things} are the numbers of all points and *things* points, respectively.

For confidence score regression, another FC layer is used to predict the per-point confidence score $\hat{c}_i$ to indicate the accuracy degree of $\hat{\boldsymbol{o}}_i$. $\hat{c}_i$ is activated by a sigmoid function to ensure $\hat{c}_i \in [0, 1]$. The ground-truth label c_i for supervising $\hat{c}_i$ is generated by

$$c_i = \begin{cases} \exp(-\frac{\mathcal{L}_{\boldsymbol{o}_i}^2}{2\sigma^2}) & \text{if } s_i \in \textit{things} \text{ classes}, \\ 0 & \text{otherwise}. \end{cases} \tag{10}$$

For the *things* points, the lower $\mathcal{L}_{\boldsymbol{o}_i}$ is, the higher c_i is. It means that the point with a more accurate regression of the center offset has a higher confidence score.

The weighted binary cross entropy loss $\mathcal{L}_{wbce}$ is applied,

$$\mathcal{L}_{wbce} = -\frac{1}{N} \sum_{i=0}^{N-1} 6c_i \log(\hat{c}_i) + (1 - c_i) \log(1 - \hat{c}_i), \tag{11}$$

where the *things* points are manually emphasized since the number of them is much less than that of the *stuff* points.

Finally, the loss for each group of results (from the CFNet or CFFE) is defined as

$$\mathcal{L} = \mathcal{L}_{wce} + 3\mathcal{L}_{ls} + \mathcal{L}_{tc} + 2\mathcal{L}_o + \mathcal{L}_{wbce}. \tag{12}$$

The total loss is the sum of the two losses sourced from the CFNet and CFFE.

3.3. Center Deduplication Module

Given the final predictions of the semantic segmentation $\hat{\boldsymbol{S}}$, center offsets $\hat{\boldsymbol{O}}$, and confidence scores $\hat{\boldsymbol{C}}$, this section introduces how to exploit the detected centers to acquire the panoptic segmentation results during inference, and illustrates the key module, center deduplication module (CDM).

Post-process. For the panoptic segmentation results, instance segmentation is first generated and then the final

Algorithm 1: Center deduplication module

```
Input: Shifted things points $\{\tilde{\boldsymbol{p}}_t^{shift} \in \mathbb{R}^3\}_{t=0}^{M-1}$,
       confidence scores $\{\tilde{c}_t \in [0,1]\}_{t=0}^{M-1}$,
       distance threshold $d \in \mathbb{R}$
Output: Instance centers $\{\tilde{\boldsymbol{e}}_j \in \mathbb{R}^3\}_{j=0}^{D-1}$
$order$ = sort_index($\{\tilde{c}_t\}_{t=0}^{M-1}$, decending=true)
initialize a boolean vector $rej \in \mathbb{B}^M$ with all
  elements of false and an empty list $keep$
for $k \leftarrow 0$ to $M-1$ do
    if not $rej[order[k]]$ then
        $keep.append(order[k])$
        for $h \leftarrow k+1$ to $M-1$ do
            $dist = ||\tilde{\boldsymbol{p}}_{order[k]}^{shift} - \tilde{\boldsymbol{p}}_{order[h]}^{shift}||_2$
            if $dist < d$ then
                $rej[order[h]] =$ true
            end
        end
    end
end
$\{\tilde{\boldsymbol{e}}_j\} = \tilde{\boldsymbol{p}}_{keep}^{shift}$
```

panoptic segmentation is acquired by fusing the semantic and instance segmentation labels. There are five steps to achieve the final panoptic segmentation:

1) The *things* points $\{\tilde{\boldsymbol{p}}_t\}_{t=0}^{M-1} = \{\boldsymbol{p}_i | \hat{s}_i \in$ *things* classes$\}$ are selected according to the predicted semantic labels, with their offsets $\{\tilde{\boldsymbol{o}}_t\}_{t=0}^{M-1}$ and confidence scores $\{\tilde{c}_t\}_{t=0}^{M-1}$ (where M is the number of *things* points).

2) Each shifted *things* point $\tilde{\boldsymbol{p}}_t^{shift}$ is given by $\tilde{\boldsymbol{p}}_t^{shift} = \tilde{\boldsymbol{p}}_t + \tilde{\boldsymbol{o}}_t$ and is severing as an instance center candidate.

3) A center $\tilde{\boldsymbol{e}}_j$ for each instance j is selected by the CDM based on the coordinates $\tilde{\boldsymbol{p}}_t^{shift}$ and its confidence score $\tilde{c}_t$, while other candidates are suppressed.

4) The instance ID $\widehat{inst}_t$ is acquired by assigning the shifted *things* points $\tilde{\boldsymbol{p}}_t^{shift}$ to the closest center in all centers $\{\tilde{\boldsymbol{e}}_j\}_{j=0}^{D-1}$ (D is the number of detected instances).

5) The majority voting [15, 39] reassigns the most frequent semantic label to all points of a predicted instance to further guarantee the consistency of semantic labels within a predicted instance.

Center deduplication module (CDM). The CDM takes the shifted points $\{\tilde{\boldsymbol{p}}_t^{shift}\}_{t=0}^{M-1}$ and confidence scores $\{\tilde{c}_t\}_{t=0}^{M-1}$ as input to get an instance center $\hat{\boldsymbol{e}}_j$ for each instance j. Inspired by the bounding box NMS, our CDM suppresses the candidate centers with lower scores within a euclidean distance threshold d. The pseudo-code of our CDM is shown in Algorithm 1, where two centers with a distance less than d are considered as the same instance. The process of our CDM is simple and can be easily implemented in CUDA.

4. Experiments

The proposed CFNet is compared with the existing methods on the public SemanticKITTI [2] and nuScenes [5] panoptic segmentation benchmarks.

4.1. Experimental Setup

Datasets. The SemanticKITTI [2] consists of 22 sequences collected by a Velodyne HDL-64E 360° rotating LiDAR with 64 beams vertically. It contains 43,552 LiDAR scans in total and is split into a training set with 19,130 scans from sequences 00 to 10 except 08, a validation set containing sequence 08 with 4,071 scans, and a test set including the rest sequences with 20,351 scans. The test set is only provided with point clouds and used for the online leaderboards. For the panoptic segmentation, it has 19 valid categories, including 11 *stuff* and 8 *things* categories.

The nuScenes [5] is a newly released benchmark with 1,000 scenes collected by a Velodyne HDL-32E 360° rotating LiDAR with 32 beams vertically. The dataset is collected in Boston and Singapore. It uses 28,130 scans for training, 6,019 for validation, and 6,008 for testing. The panoptic segmentation benchmark provides 16 categories, including 6 *stuff* and 10 *things* categories [12].

Evaluation metric. As the official benchmarks [3, 12] suggest, the panoptic quality (PQ) is adopted to evaluate the performance of LiDAR panoptic segmentation. The PQ [3, 12] measures the overall quality of the panoptic segmentation, and can be decomposed into two terms: 1) the segmentation quality (SQ) is the average IoU of all matched pairs; 2) the recognition quality (RQ) measures the F_1 score. These three metrics are also reported separately on the *stuff* and *things* classes, including PQSt, SQSt, RQSt, and PQTh, SQTh, RQTh, respectively. The PQ† [26] is also reported by replacing the PQ of each *stuff* class with its IoU. For evaluating the sub-task of semantic segmentation, mean Intersection over Union (mIoU) [2] is reported.

Training details. In the experiments, two 2D projection-based backbones, namely PolarNet [38] and CPGNet [23], are used in our CFNet. For the backbone of PolarNet [38], the training schedules and data augmentation are the same as those of Panoptic-PolarNet [39] for a fair comparison. Besides, for efficiency, only one stage version of CPGNet [23] is used in our CFNet. It is trained from scratch for 48 epochs with a batch size of 8 and takes around 24 hours on 8 NVIDIA RTX 3090 GPUs. Stochastic gradient descent (SGD) is used as the optimizer, where the initial learning rate, weight decay, and momentum are 0.02, 0.001, and 0.9, respectively. The learning rate is decayed by 0.1 per 10 epochs. In training, the data augmentation is applied as a convention, including random flipping along the x and y axes, random global scale sampled from $[0.95, 1.05]$, random rotation around the z-axis, and random Gaussian noise $\mathcal{N}(0, 0.02)$.

	Backbone	CFFE		Deduplication	PQ	PQ†	PQTh	PQSt	mIoU	RT(ms)
		CFG	FEM							
a	PolarNet [38]			BEV Center Heatmap [39]	59.1	64.1	65.7	54.3	64.5	65.3 + 20.9
b			✓	BEV Center Heatmap [39]	59.5	64.3	66.2	54.4	64.6	69.7 + 20.9
c		✓	✓	BEV Center Heatmap [39]	60.4	65.6	67.2	55.1	65.2	71.9 + 20.9
d		✓	✓	CDM [Ours]	**60.6**	**65.7**	**67.8**	**55.2**	**65.4**	**71.9 + 3.6**
e	CPGNet [23]			DBSCAN [11]	59.5	64.1	63.8	56.3	64.6	31.6+24.9
f				HDBSCAN [6]	58.3	62.9	60.9	56.3	64.9	31.6+48.7
g				MeanShift [9]	59.9	64.5	64.7	56.3	64.9	31.6+84.6
h				CDM [Ours]	60.5	65.6	66.1	56.5	66.3	**32.7+2.3**
i			✓	CDM [Ours]	60.8	65.9	66.3	56.9	66.5	37.4+2.3
j		✓	✓	CDM [Ours]	**62.7**	**67.5**	**70.0**	**57.3**	**67.4**	41.2+2.3

Table 1. Ablation studies on the SemanticKITTI validation set. RT: running time.

4.2. Ablation Studies

Ablative analyses are conducted on the SemanticKITTI validation set with the above experimental setup. As shown in Table 1, two 2D projection-based backbones, namely PolarNet [38] and CPGNet [23], are incorporated with our framework to validate the proposed CFFE and CDM. Row (a) shows the original Panoptic-Polarnet [39], while rows (b,c,d) are implemented based on the official code of Panoptic-Polarnet [39]. The running time is reported by the time of the model inference plus that of the post-process.

Effects of the CFFE. The CFFE is configured in three ways: 1) the CFFE is not applied (a,e,f,g,h); 2) the center feature generation (CFG) in the CFFE is removed (b,i), which is equal to the setting of only imposing the intermediate supervision; 3) the whole CFFE is applied (c,d,j). For the two backbones, a common observation is that the improvements primarily come from the CFG rather than the intermediate supervision, especially for the *things* classes (a,b,c and h,i,j). In terms of the running time, The CFFE brings acceptable latency of +6.6 ms and +8.5 ms for these two backbones, respectively.

Moreover, the errors in predicting *things* center offsets are calculated for both the CFNet with CFFE and its intermediate results as shown in Table 2. Equipped with the backbone of CPGNet [23], the proposed CFFE reduces the errors of predicting the *things* center offsets. The observations demonstrate that explicitly modeling center features helps in identifying an instance.

Effects of the CDM. For the proposal-free methods, there are primarily three ways to identify each instance: 1) the class-agnostic clustering methods (*e.g*. DBSCAN, HDBSCAN, MeanShift) (e,f,g), 2) the BEV center heatmap [39] (a,b,c), and 3) our CDM (d,h,i,j). For the backbone of PolarNet [38], our CDM outperforms the BEV center heatmap [39] by +0.2 for the PQ (c,d), while our CDM runs faster. For the backbone of CPGNet [23], compared with the class-agnostic clustering methods, our CDM also achieves the best performance on both accuracy (+0.6∼+2.2 for the PQ) and running time (-21.5 ms∼-81.2 ms), though it slows down the model inference by +1.1 ms (g,h) owing to the additional confidence prediction.

More ablation studies can be seen in the supplementary material.

	training set	validation set
Intermediate	1.35	1.22
CFNet	**1.19**	**1.13**

Table 2. Mean errors measured in meter (m) of *things* center offsets for the intermediate results and the CFNet with the CFFE on the SemanticKITTI training and validation set.

4.3. Evaluation Comparisons

Quantitative results. In Table 3 and Table 4, our CFNet with CPGNet as its backbone is compared with the existing methods on the SemanticKITTI test set and nuScenes validation set, respectively. From top to bottom, the methods are grouped as proposal-based and proposal-free methods. It is found that our proposal-free CFNet surpasses the previous methods by a large margin on most metrics. Specifically, on the SemanticKITTI test set, our CFNet outperforms the previous state-of-the-art SCAN [35] and Panoptic-PHNet [22] by +1.9 for the PQ, and +1.5 for the RQ. On the nuScenes validation set, our CFNet outperforms the most competitive Panoptic-PHNet [22] by +0.4 for the PQ, +0.6 for the SQ, and +0.4 for the RQ, respectively.

Running time. The running time is reported on the SemanticKITTI. It is measured with PyTorch FP32 without any model acceleration tricks on a single NVIDIA RTX 3090 GPU. For clarity, the running time of our method is further divided into two parts, including the time costs of the model and post-process. Based on the same kind of devices as the existing methods, our CFNet runs 43.5 ms (41.2 ms for model inference and 2.3 ms for post-process) and is 1.6 times faster than the most efficient Panoptic-PHNet [22].

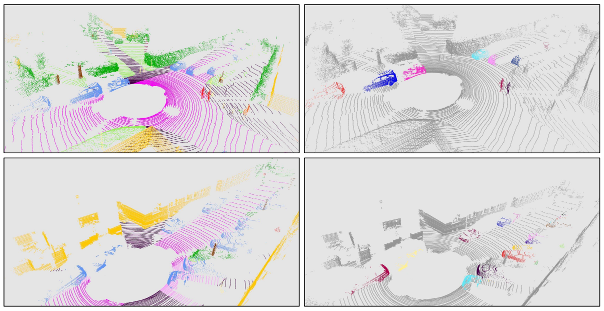
Qualitative results. The visualization results are

Methods	PQ	PQ†	SQ	RQ	PQTh	SQTh	RQTh	PQSt	SQSt	RQSt	mIoU	RT(ms)
Proposal-based Methods												
RangeNet++ [25]&PointPillars [19]	37.1	45.9	75.9	47.0	20.2	75.2	25.2	49.3	76.5	62.8	52.4	416.6
KPConv [33]&PointPillars [19]	44.5	52.5	80.0	54.4	32.7	81.5	38.7	53.1	79.0	65.9	58.8	526.3
PanopticTrackNet [17]	43.1	50.7	78.8	53.9	28.6	80.4	35.5	53.6	77.7	67.3	52.6	147
EfficientLPS [31]	57.4	63.2	83.0	68.7	53.1	87.8	60.5	60.5	79.5	**74.6**	61.4	-
Proposal-free Methods												
LPSAD [24]	38.0	47.0	76.5	48.2	25.6	76.8	31.8	47.1	76.2	60.1	50.9	84.7
DS-Net [15]	55.9	62.5	82.3	66.7	55.1	87.2	62.8	56.5	78.7	69.5	61.6	294.1
Panoster [13]	52.7	59.9	80.7	64.1	49.4	83.3	58.5	55.1	78.8	68.2	59.9	-
GP-S3Net [29]	60.0	69.0	82.0	72.1	65.0	86.6	**74.5**	56.4	78.7	70.4	**70.8**	270.3
SMAC-Seg [20]	56.1	62.5	82.0	67.9	53.0	85.6	61.8	58.4	79.3	72.3	63.3	99
Panoptic-PolarNet [39]	54.1	60.7	81.4	65.0	53.3	87.2	60.6	54.8	77.2	68.1	59.5	86.2
SCAN [35]	61.5	67.5	84.5	72.1	61.4	88.1	69.3	**61.5**	81.8	74.1	67.7	78.1
Panoptic-PHNet [22]	61.5	67.9	84.8	72.1	63.8	**90.7**	70.4	59.9	80.5	73.3	66.0	69.3
CFNet [Ours] w/CPGNet [23]	**63.4**	**69.7**	**85.2**	**73.6**	**66.7**	89.8	74.3	61.0	**81.9**	73.1	68.3	**41.2 + 2.3**

Table 3. Comparison results on the SemanticKITTI test set.

Methods	PQ	PQ†	SQ	RQ	PQTh	SQTh	RQTh	PQSt	SQSt	RQSt	mIoU
Proposal-based Methods											
Cylinder3D [40]&PointPillars [19]	36.0	44.5	83.3	43.0	23.3	83.7	27.0	57.2	82.7	69.6	52.3
Cylinder3D [40]&SECOND [36]	40.1	48.4	84.2	47.3	29.0	84.4	33.6	58.5	83.7	70.1	58.5
PanopticTrackNet [17]	51.4	56.2	80.2	63.3	45.8	81.4	55.9	60.4	78.3	75.5	58.0
EfficientLPS [31]	62.0	65.6	83.4	73.9	56.8	83.2	68.0	70.6	83.8	83.6	65.6
Proposal-free Methods											
LPSAD [24]	50.4	57.7	79.4	62.4	43.2	80.2	53.2	57.5	78.5	71.7	62.5
DS-Net [15]	42.5	51.0	83.6	50.3	32.5	83.1	38.3	59.2	84.4	70.3	70.7
GP-S3Net [29]	61.0	67.5	84.1	72.0	56.0	85.3	65.2	66.0	82.9	78.7	75.8
SMAC-Seg [20]	67.0	71.8	85.0	78.2	65.2	87.1	74.2	68.8	82.9	82.2	72.2
Panoptic-PolarNet [39]	63.4	67.2	83.9	75.3	59.2	84.1	70.3	70.4	83.6	83.5	66.9
SCAN [35]	65.1	68.9	85.7	75.3	60.6	85.7	70.2	72.5	85.7	83.8	77.4
Panoptic-PHNet [22]	74.7	77.7	88.2	84.2	74.0	89.0	82.5	75.9	86.8	86.9	**79.7**
CFNet [Ours] w/CPGNet [23]	**75.1**	**78.0**	**88.8**	**84.6**	**74.8**	**89.8**	**82.9**	**76.6**	**87.1**	**87.3**	79.3

Table 4. Comparison results on the nuScenes validation set.



Semantic Segmentation

Instance Segmentation

Figure 5. Visualizations of our CFNet on the SemanticKITTI test set. Different colors represent different classes or instances.

shown in Fig. 5. Our CFNet can distinguish adjacent pedestrians or cars. Besides, the boundaries of instances can be accurately segmented. More qualitative results can be seen in the supplementary material.

5. Conclusion

A novel proposal-free CFNet is proposed for real-time LiDAR panoptic segmentation. For better modeling and exploiting the non-existent instance centers, a novel CFFE is proposed to generate enhanced center-focusing feature maps and a CDM is introduced to keep only one center for each instance and then assign the shifted *things* points to the closest center for acquiring instance IDs. From the experiments, it can be seen that center modeling and exploiting is a key problem in the proposal-free LiDAR panoptic segmentation methods, and mimicking the non-existent center features is promising and shows a clear benefit.

Acknowledgements

The research project is partially supported by the National Key R&D Program of China (No.2021ZD0111902), and the National Natural Science Foundation of China (No.U21B2038, 62272015, U19B2039).

References

[1] Yara Ali Alnaggar, Mohamed Afifi, Karim Amer, and Mohamed ElHelw. Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1800–1809, 2021. 2

[2] 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

[3] Jens Behley, Andres Milioto, and Cyrill Stachniss. A benchmark for lidar-based panoptic segmentation based on kitti. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13596–13603. IEEE, 2021. 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. 5

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 6

[6] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. 7

[7] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12547–12556, 2021. 2

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2

[9] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 3, 7

[10] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing*, pages 207–222. Springer, 2020. 2, 4

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. 3, 7

[12] 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. 6

[13] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. Panoster: End-to-end panoptic segmentation of lidar point clouds. *IEEE Robotics and Automation Letters*, 6(2):3216–3223, 2021. 1, 3, 8

[14] 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. 1, 2

[15] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13090–13099, 2021. 1, 3, 5, 6, 8

[16] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 2

[17] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189*, 2020. 1, 2, 8

[18] 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

[19] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1, 2, 4, 8

[20] Enxu Li, Ryan Razani, Yixuan Xu, and Liu Bingbing. Smac-seg: Lidar panoptic segmentation via sparse multi-directional attention clustering. *arXiv preprint arXiv:2108.13588*, 2021. 2, 3, 8

[21] Enxu Li, Ryan Razani, Yixuan Xu, and Bingbing Liu. Cpseg: Cluster-free panoptic segmentation of 3d lidar point clouds. *arXiv preprint arXiv:2111.01723*, 2021. 1

[22] 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. 1, 3, 7, 8

[23] Xiaoyan Li, Gang Zhang, Hongyu Pan, and Zhenhua Wang. Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation. *arXiv preprint arXiv:2204.09914*, 2022. 2, 4, 5, 6, 7, 8

[24] 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, 8

[25] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. 2, 4, 8

[26] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8277–8286, 2019. 6

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[28] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[29] 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. 1, 3, 8

[30] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1

[31] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, 2021. 1, 2, 8

[32] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, pages 685–702. Springer, 2020. 2

[33] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 2, 8

[34] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020. 2, 4

[35] Shuangjie Xu, Rui Wan, Maosheng Ye, Xiaoyi Zou, and Tongyi Cao. Sparse cross-scale attention network for efficient lidar panoptic segmentation. *arXiv preprint arXiv:2201.05972*, 2022. 1, 2, 3, 7, 8

[36] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 8

[37] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *arXiv preprint arXiv:2209.09385*, 2022. 1, 2

[38] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 2, 4, 6, 7

[39] 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. 1, 2, 3, 6, 7, 8

[40] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021. 2, 8