CenterLPS: Segment Instances by Centers for LiDAR Panoptic Segmentation

Jianbiao Mei*
jianbiaomei@zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

Yu Yang*
yu.yang@zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

Mengmeng Wang
mengmengwang@zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

Zizhang Li
zzli@zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

Xiaojun Hou
houxiaojun@zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

Jongwon Ra
jongwonra@zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

Laijian Li
lilaijian@zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

Yong Liu†
yongliu@iipc.zju.edu.cn
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China

ABSTRACT

This paper focuses on LiDAR Panoptic Segmentation (LPS), which has attracted more attention recently due to its broad application prospect for autonomous driving and robotics. The mainstream LPS approaches either adopt a top-down strategy relying on 3D object detectors to discover instances or utilize time-consuming heuristic clustering algorithms to group instances in a bottom-up manner. Inspired by the center representation and kernel-based segmentation, we propose a new detection-free and clustering-free framework called CenterLPS, with the center-based instance encoding and decoding paradigm. Specifically, we propose a sparse center proposal network to generate the sparse 3D instance centers, as well as center feature embedding, which can well encode characteristics of instances. Then a center-aware transformer is applied to collect the context between different center feature embedding and around centers. Moreover, we generate the kernel weights based on the enhanced center feature embedding and initialize dynamic convolutions to decode the final instance masks. Finally, a mask fusion module is devised to unify the semantic and instance predictions and improve the panoptic quality. Extensive experiments on SemanticKITTI and nuScenes demonstrate the effectiveness of our proposed center-based framework CenterLPS.

CCS CONCEPTS

• **Computing methodologies → Scene understanding**.

KEYWORDS

LiDAR panoptic segmentation; Sparse center proposal network; Center-aware transformer; Mask fusion

ACM Reference Format:
Jianbiao Mei, Yu Yang, Mengmeng Wang, Zizhang Li, Xiaojun Hou, Jongwon Ra, Laijian Li, and Yong Liu. 2023. CenterLPS: Segment Instances by Centers for LiDAR Panoptic Segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3581783.3612080

1 INTRODUCTION

LiDAR is an essential tool for sensing and perception in autonomous driving and robotics, providing highly accurate 3D point cloud data of the environment. Typically, LiDAR segmentation aims to predict point-level segmentation, allowing for a more comprehensive understanding of the 3D scene. This paper focuses on LiDAR Panoptic Segmentation (LPS), a prevalent 3D scene understanding problem. LPS unifies semantic and instance segmentation tasks, assigning semantic categories and instance IDs for each point in the LiDAR point cloud. It requires parsing the *stuff* (*e.g.*, road, building, and vegetation) and identifying the *thing* (*e.g.*, car, cyclist, and person).

LiDAR point clouds can be sparse, noisy, and occluded, making it difficult to provide accurate segmentation and distinguish between different instances. Despite these challenges, recent advances in deep learning have led to significant progress in LPS. To obtain reliable LiDAR panoptic segmentation, one of the critical problems is accurately localizing and segmenting instances. Regarding the implementation of instance segmentation, most existing methods follow two directions, *i.e.*, detection-based and clustering-based, to

[*] Equal contribution
[†] Corresponding author

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612080

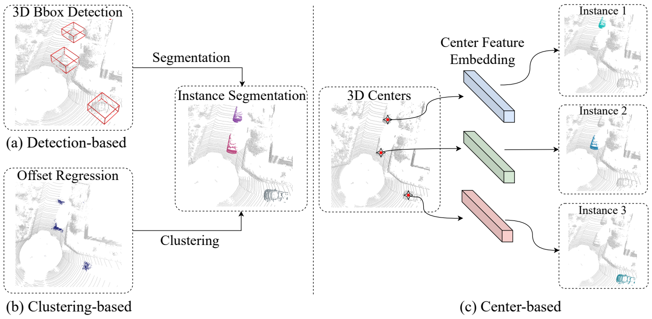

Figure 1: The mainstream LPS methods, *i.e.*, Detection-based (a), Clustering-based (b) and our center-based framework (c).

address these challenges. Detection-based methods [15, 31, 39, 40] adopt a top-down strategy, which depends on object detection as an independent branch to predict the region proposals and perform instance segmentation based on proposals (Fig. 1 (a)). However, these methods heavily rely on the detector's performance, and the generation procedure of region proposals for object localization involves a large computational overhead. On the other way, clustering-based methods [7, 9, 14, 19, 20, 27, 30, 46] utilize the geometric shifts predicted by an offset branch to implicitly localize instances and take heuristic clustering algorithms to group instances in a bottom-up manner (Fig. 1 (b)). While the process of heuristic instance grouping is usually time-consuming. Besides, they usually introduce many hand-crafted hyper-parameters, which are sensitive and limit the robustness, and may lead to over-segmented problems in practice. Overall, existing LPS methods suffer from sub-optimal performance caused by either object detection or clustering algorithms.

Motivated by the center representation [43, 45, 47] in 2D/3D object detection, we propose a new detection-free and clustering-free framework (Fig. 1 (c)) for LiDAR panoptic segmentation, termed as **CenterLPS**, which localizes and segments instances by centers to solve the challenges above. We aim to predict the 3D centers to localize instances and encode instances into center feature embedding. Then the instance masks can be further decoded via center feature embedding. The main insight lies that the instance centers and feature embedding encode the location and object characteristics, which can be used to represent each object and distinguish between different instances. Specifically, we develop a Sparse Center Proposal Network (SCPN) to generate 3D instance centers based on the pseudo heatmap [20]. SCPN projects the shifted *thing* points onto a BEV image to generate the 2D pseudo heatmap utilized to find the pillar that 3D instance centers belong to by window-based max-pooling. Furthermore, SCPN generates the 3D centers and center feature embedding based on points in the selected pillars. The semantic categories of these 3D centers are directly assigned by majority voting on points in the selected pillars to explicitly decouple the classification and segmentation to mitigate the competition between them, as proven in [12]. Compared to dense 2D/3D heatmap learning for centers as [36, 43, 45, 47], our SCPN has two key advantages: (1) Sparsity. There is no necessity to adopt top-k along with NMS operation for center filtering. (2) Flexibility and scalability. The maximum number of instances no longer needs to be set in advance. SCPN is scalable to handle a large number of objects in the urban scene.

Moreover, the abundant contextual information is beneficial for instance encoding. Therefore, we design the center-aware transformer to collect context between different center feature embedding and around the centers. By modeling the inter- and intra-instance dependencies with global self-attention and local cross-attention, we enhance the center feature embedding with more informative cues, facilitating subsequent instance decoding.

Furthermore, we find that directly predicting instance masks based on the class-known center feature embedding can naturally avoid semantic conflicts in the same cluster brought by class-agnostic clustering on *thing* points and correct potential wrong semantic predictions. On the other hand, kernel-based methods [4–6, 22, 35, 44] have recently achieved wide success in the 2D segmentation. However, there are few studies to explore kernel-based LiDAR panoptic segmentation. Thus, we utilize dynamic convolution as [13, 35, 36] to generate kernel weights and initialize a few convolution layers that are used to decode the final instance masks. The dynamic convolution is conditional on the enhanced center feature embedding, implying object-aware information such as instance location, shape, and size. To expedite the convergence of the network, we further enhance the dynamic convolution with position and shape priors. Moreover, observing that multiple centers may be generated for a single instance (especially large objects) due to inaccurate offsets prediction, we design a mask fusion module to merge the masks decoded by the centers belonging to the same instance and paste the merged masks on the semantic predictions to unify the semantic predictions and instance masks. Our mask fusion module merges the potentially overlapped masks belonging to the same instance and improves the panoptic quality.

We evaluate our CenterLPS on two large-scale outdoor datasets SemanticKITTI [1] and nuScenes [8]. Extensive experiments demonstrate the effectiveness of our method.

The main contributions of this paper are summarized as follows:

- We propose a new detection-free and clustering-free framework, dubbed as **CenterLPS**, with the paradigm of center-based instance encoding and decoding for LiDAR panoptic segmentation.
- We develop a sparse center proposal network based on the pseudo heatmap to predict instance centers and feature embedding, which can well capture object information of instances.
- A center-aware transformer is designed to collect context between different center feature embedding and around centers. The center-based queries facilitate the learning of the transformer.
- We introduce dynamic convolution with position/shape priors to decode instance masks. A mask fusion module is devised to unify the semantic and instance predictions.

2 RELATED WORKS

2.1 LiDAR Panoptic Segmentation

In terms of the implementation of instance segmentation, most existing LPS methods can be divided into two types of frameworks, *i.e.*, detection-based and clustering-based methods.

Detection-based methods [15, 31, 39, 40] use an independent 3D detection branch to predict object region proposals and segment instances based on proposals. SemanticKITTI [2] and nuScenes [8]

released LiDAR panoptic segmentation datasets, exploring this task with joint object detectors and semantic segmentation networks. PanopticTrackNet [15] follows Mask R-CNN [12], utilizing a regional proposal network (RPN), bounding box regression, and mask generation for instance segmentation. Recently, LidarMultiNet [40] proposed a multi-task network unifying 3D object detection and segmentation, predicting refined panoptic segmentation by fusing detection and semantic segmentation results. However, these methods heavily rely on detected region proposals, incurring additional computational consumption during proposal generation.
Clustering-based methods [7, 9, 14, 19–21, 26, 27, 30, 46] implicitly localize instances by predicting offset vectors or embedding vectors for *thing* points, and then apply heuristic clustering algorithms to group instances. DS-Net [14] designed a learnable dynamic shift module that iteratively regresses centers for subsequent clustering. Panoptic-PolarNet [46] predicts the center heatmap and performs clustering among the shifted points on the polar BEV map. SMAC-Seg [19] presented a novel multi-directional attention clustering module to segment multi-scale instances. Panoptic-PHNet [20] introduced a pseudo heatmap generated from the shifted *thing* points and a center grouping module to yield 2D instance centers for efficient clustering. However, these clustering-based methods are either time-consuming or sensitive to hyper-parameters.

Unlike these mainstream LPS methods, our CenterLPS eliminates the dependence on object detection and clustering algorithms for instance localization and segmentation. It adopts the paradigm of center-based instance encoding and decoding to localize and segment instances effectively and efficiently, which can exert the power of the center and kernel representations.

2.2 Kernel-based Segmentation

Recently, kernel-based 2D segmentation methods [5, 6, 22, 25, 35, 44] have been studied extensively. CondInst [35] encodes instances into dynamic filters for decoding instances. Panoptic-FCN [22] implements 2D panoptic segmentation using kernel generators in a unified workflow. K-Net [44] designs a kernel update strategy for consistent instance and semantic segmentation. MaskFormer [6] uses the transformer with learnable queries to output binary masks with class labels for semantic segmentation. Mask2Former [5] applies masked attention for universal image segmentation.

Building on 2D segmentation, there are kernel-based studies [13, 32, 36, 37] for 3D segmentation tasks. DyCo3D [13] generates convolution filters based on instances to decode instance masks. DKNet [36] represents instances as kernels encoding semantic, positional, and shape information of 3D instances. Following MaskFormer, MaskRange [11], and MaskPLS [24] use learnable queries for range-based and point-based LiDAR panoptic segmentation, respectively. PUPS [32] employs point-level classifiers to predict semantic masks and instance groups directly.

Similar to kernel-based methods, our CenterLPS uses kernel representation and dynamic convolution to generate conditional kernel weights from the center feature embedding for effective instance mask decoding. This naturally avoids semantic conflicts from class-agnostic clustering on *thing* points. We also enhance dynamic convolution with position and shape priors for faster network convergence.

2.3 Center Representation

In the domain of 2D object detection, CenterNet [45] first proposes to model the 2D object as a single point, *i.e.*, the center point of its bounding box, which bypasses the need for anchor boxes. For 3D object detection, CenterPoint [43] proposed to represent, detect, and track 3D objects as points. Following CenterNet, CenterPoint detects centers of objects using a 2D center heatmap head and regresses to other attributes, including 3D size, 3D orientation, and velocity. By employing the center representation, CenterPoint realizes simple yet efficient, real-time, and accurate detection performance. Based on CenterPoint, CenterFormer [47], a query-based and center-based 3D detection framework, introduces the transformer with the center embedding as queries to extract object features and predict the bounding box effectively. On the other hand, the very recent 3D instance segmentation method DKNet [36] proposes to localize instances by a learned 3D centroid heatmap and devises an aggregation strategy to merge duplicate candidates.

As proven by these pioneer works, instance centers and feature embedding encode the location and object characteristics, which can well represent each object and distinguish between different instances. Thus, we adopt the center-based instance encoding and decoding paradigm to localize and segment instances for outdoor LiDAR panoptic segmentation. However, different from these methods, which learn a dense 2D/3D Gaussian heatmap for center discovery and exploit top-k strategy for center filtering, our CenterLPS predicts the sparse 3D instance centers and feature embedding based on a pseudo heatmap, which does not require fixing the maximum number of instances and is simple and flexible.

3 METHOD

3.1 Overview

In this paper, we propose to segment instances by centers for LiDAR panoptic segmentation. The overall framework is illustrated in Fig. 2. Firstly, the voxel-based backbone, *i.e.*, GASN [41], is applied to extract the point-wise features, and a two-layer MLP is used to provide the semantic prediction (Sec. 3.2). Then, we devise the sparse center proposal network to generate sparse 3D centers as well as the center feature embedding (Sec. 3.3). And a center-aware transformer is designed to collect contextual information to enhance the center feature embedding by modeling the inter- and intra-instance relationships (Sec. 3.4). Finally, dynamic convolution is utilized to decode the instance masks; and a mask fusion module is developed to unify the semantic and instance prediction and improve the panoptic quality (Sec. 3.5).

3.2 Point-wise Semantic Prediction

Following GASN [41], we take the multi-scale sparse 3D CNN as the backbone to aggregate multi-scale 3D features. Given the input LiDAR point cloud $P \in \mathbb{R}^{N \times 4}$ (coordinates and intensity), a voxelization layer similar to DRINet [42] is utilized to obtain the voxel-wise features F_v^0 with a dense spatial resolution of $L \times H \times W$. After that, four cascaded encoder blocks used in GASN extract the multi-scale 3D features $(F_v^1, F_v^2, F_v^3, F_v^4)$. And these voxel-wise 3D features are further back-projected to get the point-wise features $(F_p^0, F_p^1, F_p^2, F_p^3, F_p^4)$, which imply multi-scale contextual and

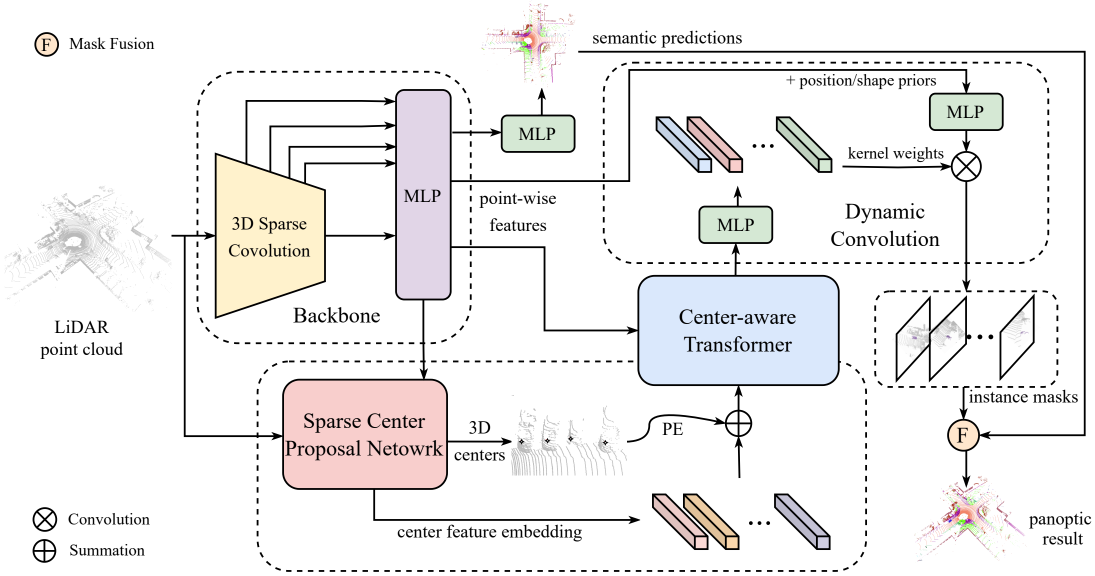

Figure 2: Pipeline of our CenterLPS. The framework consists of a voxel-based backbone for point-wise features, a semantic head for semantic predictions, a sparse center proposal network for instance encoding, a center-aware transformer for collecting context, dynamic convolutions for instance decoding, and a mask fusion module to unify the semantic and instance predictions.

geometric information. After that, these point-wise features are concatenated along channel dimension and fed into the linear layer for encoding the final point features $F_p \in \mathbb{R}^{N \times D}$. Finally, a semantic head consisting of a two-layer MLP takes point features F_p and outputs the semantic scores $S = \{s_1, ..., s_N\} \in \mathbb{R}^{N \times N_{class}}$ for N points over N_{class} categories. During training, we use cross-entropy loss and lovasz loss [3] to supervise the semantic backbone.

3.3 Sparse Center Proposal Network

After the point-wise semantic prediction, we take the remaining work as the instance segmentation task. And different from the mainstream detection- and clustering-based methods, inspired by center representation [36, 43, 45, 47], we propose to segment instances by centers. Specifically, a sparse center proposal network is devised to locate and encode instances, as illustrated in Fig. 3. To begin with, we construct a two-layer MLP to learn the offset vectors $O = \{o_1, ..., o_N\} \in \mathbb{R}^{N \times 3}$, representing the geometric shifts from each point to the center of the instance that it belongs to. We use $L1$ regression loss and cosine direction loss to construct the $\mathcal{L}_{off}$ to optimize the shift vectors during training:

$$\mathcal{L}_{off} = \frac{1}{\sum_i m_i} \sum_i (|o_i - (\hat{c}_i - p_i)| + \frac{o_i \cdot (\hat{c}_i - p_i)}{\|o_i\| \cdot \|\hat{c}_i - p_i\|}) \cdot m_i \quad (1)$$

where $M = \{m_1, ..., m_N\}$ is a binary mask, indicating the *thing* points. m_i is set 1 if point i belongs to *thing* points, and otherwise 0. $\hat{c}_i$ is the instance center that point i lies in.

With the predicted offset vectors, the *thing* points are first shifted toward instance centers to make points in the same instance closer to each other. Then, we project the shifted *thing* points onto a BEV image to generate a 2D pseudo heatmap [20] based on the assumption that the geometric centers of outdoor instances are separate from each other under the bird's eye view. Specifically, we first discrete the shifted *thing* points into an evenly spaced grid in the x-y plane to create a set of pillars P with $|P| = H \cdot W$ as [17]. And each grid I_i in the BEV image $I \in \mathbb{R}^{H \times W}$ corresponds with a pillar P_i. Then we can acquire the quantitative density of the pseudo heatmap by counting the number of points in the pillar. Furthermore, we leverage window-based max pooling to pick out the local maximum, which is used to select the pillar where the 3D instance center may exist. By scattering the coordinates and features of points in the k-th selected pillar P_k, the 3D center c_k and center feature embedding $f_{c_k} \in \mathbb{R}^D$ can be generated. The scatter procedure is achieved by:

$$c_k = \text{AvgPool}(G_k^p) \quad (2)$$

$$f_{c_k} = \text{MLP}[\text{AvgPool}(G_k^f)] \quad (3)$$

where G_k^p and G_k^f are points and point features in the pillar P_k. As proven in Mask R-CNN [12], explicitly decoupling segmentation and classification can alleviate the competition problem between them. Thus, we take the majority voting strategy based on the semantic prediction of the points in the corresponding pillar to directly assign the semantic category to the center feature embedding. Different from the commonly used heatmap learning for centers [36, 47], our SCPN does not require fixing the maximum number of instances in advance and is scalable to handle a large number of objects in the urban scene. We provide visualizations of the sparse 3D centers generated by our SCPN in Fig.4. The results show that our SCPN generates accurate centers in complex scenes and can handle crowded scenes well. The cases where multiple centers may be generated for large objects are addressed by our mask fusion module, which will be explained in Sec. 3.5.

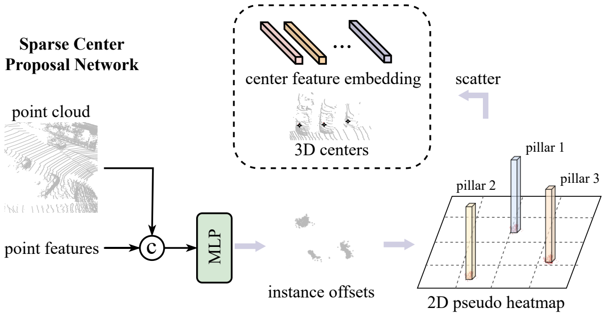

Figure 3: The sparse center proposal network.

3.4 Center-aware Transformer

Let $P_c = \{c_1, ..., c_{N_c}\} \in \mathbb{R}^{N_c \times 3}$ and $F_c = \{f_{c_1}, ..., f_{c_{N_c}}\} \in \mathbb{R}^{N_c \times D}$ denote 3D centers and center feature embedding generated by the sparse center proposal network for N_c instances. We design the center-aware transformer to collect the context between different center feature embedding and around the centers and output the enhanced center feature embedding $F'_c = \{f'_{c_1}, ..., f'_{c_{N_c}}\}$. Specifically, we take the center feature embedding as the query features and encode the centers into the position embedding using a linear layer to inject position information explicitly. The center-aware transformer consists of L encoder blocks. Each block has a self-attention layer, a local cross-attention layer, and a feed-forward layer. A skip connection is used for each layer to connect the normalized input features using layer normalization and the output features. The self-attention layer aims to model the inter-instance dependencies. Let $F^s_c = \{f^s_{c_1}, ..., f^s_{c_{N_c}}\} \in \mathbb{R}^{N_c \times D}$ denote the input of the self-attention layer. The multi-head self-attention is achieved as:

$$MHSA(c_i) = \sum_{h=1}^{H} \phi_h [\sum_{j=1}^{N_c} \sigma(\frac{q_i k_j}{\sqrt{D}}) \cdot v_j] \tag{4}$$

$$q_i = \phi_q(f^s_{c_i}) + \delta(c_i) \tag{5}$$

$$k_j = \phi_k(f^s_{c_j}) + \delta(c_j), v_j = \phi_v(f^s_{c_j}) \tag{6}$$

where h is the head index, ϕ_* is the linear layer, and σ denotes the $softmax$ function. δ is the linear layer for position embedding. c_i is the center for $i\text{-}th$ instance.

We also apply the local cross-attention to learn the intra-instance relationships. The local cross-attention explicitly makes the query center focus on its neighbor points in the point cloud and effectively reduces the computation complexity. Similar to the voxelization operation in Sec. 3.2, we scatter the points and point features by average pooling according to the voxel indices to downsample the point cloud. Then, the indices of k nearest points are calculated for each center based on the spatial location. Through the indices, the attentive points $\Omega^p_{c_i} \in \mathbb{R}^{k \times 3}$ and the corresponding point features $\Omega^f_{c_i} \in \mathbb{R}^{k \times D}$ for center c_i are indexed. In the multi-head local cross-attention, similar to MHSA, the query q_i is calculated from the center feature embedding and position. While the key k_j and value v_j come from the attentive points $\Omega^p_{c_i}$ and point features $\Omega^f_{c_i}$. The formulation of multi-head local cross-attention is expressed as:

$$MHCA(c_i) = \sum_{h=1}^{H} \phi_h [\sum_{j \in \mathcal{N}(c_i)} \sigma(\frac{q_i k_j}{\sqrt{D}}) \cdot v_j] \tag{7}$$

where $\mathcal{N}(c_i)$ denotes the neighbors of center c_i.

3.5 Dynamic Convolution Network

After obtaining the enhanced center feature embedding as well as their semantic categories, we aim to decode the instances masks further. The clustering-based methods group instances upon the *thing* points, which usually suffer from the semantic conflicts problem in the same cluster due to the wrong semantic segmentation or inaccurate clustering. We introduce dynamic convolution based on the enhanced center feature embedding to naturally avoid the above problem by decoding the instance mask of the scene around the instance center. We first use the kernel branch consisting of a stack of convolution layers to generate the kernel weights $W_c = \{w_{c_1}, ..., w_{c_{N_c}}\} \in \mathbb{R}^{N_c \times L_w}$ from the enhanced center feature embedding generated by the center-aware transformer. The L_w denotes the dimension of the kernel weights. After that, the kernel weights, which have encoded the object characteristics of instances such as positional and shape cues, are transformed into the weights of shallow convolution network $\mathcal{F}_w$ as [13, 36]. The network $\mathcal{F}_w$ consists of two 1×1 convolution layers. A ReLU activation function follows the first layer, and the second one is attached with a *sigmoid* function to obtain the mask decoding.

Before the convolution operation, we designed the mask branch constructed with several linear layers to generate the mask features $F_m \in \mathbb{R}^{N \times D_1}$. Since positional information is essential for distinguishing between different instances, we encode center-aware position embedding into the mask features. Similar to [13], we calculate the offsets from each point to the center it belongs to. Then the offsets are concatenated with the point features F_p. Furthermore, we concatenate the coarse binary mask with them to form the final input features $F'_p \in \mathbb{R}^{N \times (D+3+1)}$, which are fed into the mask branch to obtain mask features F_m. The coarse binary mask is generated based on instance centers and the experience radius R. The value in the mask is set to 1 if the point belongs to *thing* points and is within the R-radius neighbor of a certain center, and 0 otherwise. In this way, we can inject the position and shape priors to the mask features, expediting the network's convergence.

Given the output channels $[D_2, 1]$ of convolution network $\mathcal{F}_w$, we can compute the dimension L_w of the kernel weights by:

$$L_w = \underbrace{D_1 \times D_2}_{weight} + D_2(bias) + \underbrace{D_2 \times 1}_{weight} + 1(bias) \tag{8}$$

And the instance mask M_{c_i} for center c_i is decoded by:

$$M_{c_i} = \mathcal{F}_w(F_m, w_{c_i}) \tag{9}$$

During training, the ground truth $\hat{M}_{c_i}$ for the decoded mask M_{c_i} is determined using majority voting on the ground-truth instance IDs in the pillar that c_i belongs to. We use binary cross-entropy loss and dice loss to form the segmentation loss:

$$\mathcal{L}_{ins} = \frac{1}{N_c} \sum_{i=1}^{N_c} [2 \cdot L_{BCE}(M_{c_i}, \hat{M}_{c_i}) + L_{dice}(M_{c_i}, \hat{M}_{c_i})] \tag{10}$$

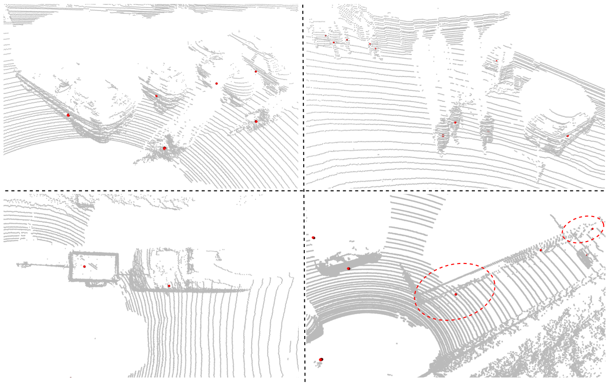

Figure 4: Our SCPN generates accurate 3D centers (red points) in complex scenes. While there still exist cases where duplicated centers (red circle) are generated (*e.g.*, large objects).

Mask Fusion. Due to inaccurate offsets prediction, multiple centers may exist in a single instance (especially large objects), as illustrated in Fig.4. Thus, we design a mask fusion module to merge the masks decoded by the centers belonging to the same instance and paste the merged masks on the semantic predictions to unify the semantic and instance masks. Specifically, let M_{c_i}, M_{c_j} be the mask decoding for center c_i, c_j with the same semantic category. We first convert the M_{c_i}, M_{c_j} to the binary masks by assigning points with a score greater than 0.5 as the foreground. Then M_{c_i} and M_{c_j} are merged if their overlap score exceeds a certain threshold θ_{thres}. The overlap score is calculated in terms of IoU between them. After the merging operation, these masks are pasted on the semantic predictions. And the semantic categories of overlapping regions are reset to be those assigned to the instance masks. The order of the pasting is based on the confidence score, defined as the average score of the foreground points. The pasting strategy can correct the possible wrong semantic segmentation and improve the final panoptic quality. Notably, due to the sparsity of the generated centers by SCPN, our mask fusion module is efficient and can be used in a plug-and-play manner. The detailed process and analysis of the mask fusion are illustrated in the appendix.

4 EXPERIMENTS

We evaluate our CenterLPS and conduct extensive experiments on the SemanticKITTI [1] and nuScenes [8] datasets. Due to the page limitation, more details on the datasets, metrics, experiments, and qualitative results are provided in the appendix.

4.1 Implementation details

We use GASN [41] as the voxel-based backbone by default. Following GASN, the voxelization resolution is $[0.2, 0.2, 0.1]$ in meters, and the voxelization space is limited in $[[\pm 48m], [\pm 48m], [-3m, 1.8m]]$. We set the model dimension $D = 64$, the channels $D_1 = 16$ for the mask feature, and output channels $D_2 = 16$ of the first layer in dynamic convolution. The number k for the center-aware transformer is set to 64. We use $L = 2$ encoder blocks for the transformer, and the number H of attention heads is set to 4. The radius R for shape prior is determined according to the average size of instances for each category in SemanticKITTI. During training, similar to [14], we apply data augmentation such as global scaling, random rotation, and random flipping on the input points of both datasets.

We combine the loss from the semantic predictions, offset prediction, and instance segmentation for the overall training loss:

$$\mathcal{L} = \mathcal{L}_{sem} + \mathcal{L}_{off} + \mathcal{L}_{ins} \tag{11}$$

where $\mathcal{L}_{sem}$ is the loss for semantic segmentation as in [41]. Our model is trained for 50 epochs following [41] for semantic segmentation, 10 epochs for the offset prediction, and another 20 epochs for instance segmentation with a total batch size of 16 on 4 NVIDIA RTX 3090 GPUs. We use the Adam [16] optimizer with an initial learning rate of $5e$-4 and a weight decay of $1e$-5 to train the network. During inference, the merging threshold θ_{thres} for the mask fusion module is set to 0.85.

4.2 Comparison with the State-of-the-art

Results on SemanticKITTI. Table 1 and Table 2 show the comparison results between our CenterLPS and other state-of-the-art methods on the SemanticKITTI validation and test sets.

- Compared with detection-based methods [15, 17, 34] our CenterLPS performs significantly better than them in terms of PQ and PQ^{Th} on both validation and test sets, demonstrating that center representation can well encode the object characteristics of instances, which can be used to represent each object and distinguish between different instances, discarding extra bounding box predictions.
- Compared with clustering-based methods [9, 14, 27, 46] and Panoptic-PHNet [20]) which employ heuristic clustering algorithms to group shifted *thing* points, our CenterLPS achieves comparable performance on validation and test sets. Notably, our CenterLPS is clustering-free and directly predicts instance mask for each center, which is robust and can naturally avoid potential semantic conflicts and correct wrong semantic predictions.
- Compared to a recent kernel-based method [24] which utilizes learnable queries to predict binary masks and semantic classes, our CenterLPS outperforms it by 2.3% and 3.4% in terms of PQ on validation and test sets, respectively. It proves that center-based queries generated by the spare center proposal network contain more effective object information, which can facilitate the learning procedure of the network. Furthermore, explicitly decoupling segmentation and classification helps alleviate mutual competition and boost the segmentation performance.

Moreover, following Panoptic-PHNet[20], we also report our version with test-time-augmentation (TTA) and model ensemble on the SemanticKITTI test set in Table 2.

Results on NuScenes. Unlike SemanticKITTI, panoptic segmentation is even more challenging on the nuScenes dataset due to the extremely sparse point clouds that are collected by a 32-beam LiDAR sensor. As shown in Table 3, our method surpasses all recent works and achieves state-of-the-art performance in terms of PQ. For example, CenterLPS exceeds SCAN [38], and Panoptic-PHNet [20] by 11.3% and 1.7%. We notice that CenterLPS surpasses these methods on *thing* classes by a large marge, e.g., 3.5% better than Panoptic-PHNet on PQ^{Th}, though it is not the best on *stuff* classes. The higher segmentation quality on *thing* classes demonstrates the effectiveness of our center-based instance encoding and decoding paradigm for LiDAR panoptic segmentation, which encodes object

Table 1: LiDAR panoptic segmentation performance on the validation set of SemanticKITTI[1]. All results in [%].

Method	PQ	PQ†	RQ	SQ	PQ$^{\mathrm{Th}}$	RQ$^{\mathrm{Th}}$	SQ$^{\mathrm{Th}}$	PQ$^{\mathrm{St}}$	RQ$^{\mathrm{St}}$	SQ$^{\mathrm{St}}$
RangeNet++[28] + PointPillars[17]	36.5	-	44.9	73.0	19.6	24.9	69.2	47.1	59.4	75.8
LPSAD[27]	36.5	46.1	-	-	-	28.2	-	-	-	-
PanopticTrackNet[15]	40.0	-	48.3	73.0	29.9	33.6	76.8	47.4	59.1	70.3
KPConv[34] + PointPillars[17]	41.1	-	50.3	74.3	28.9	33.1	69.8	50.1	62.8	**77.6**
Panoster[9]	55.6	-	66.8	79.9	56.6	65.8	-	-	-	-
Panoptic-PolarNet[46]	59.1	64.1	70.2	78.3	65.7	74.7	87.4	54.3	66.9	71.6
SCAN[38]	57.2	-	-	-	-	-	-	-	-	-
Panoptic-PHNet[20]	61.7	-	-	-	**69.3**	-	-	-	-	-
DS-Net[14]	57.7	63.4	68.0	77.6	61.8	68.8	78.2	54.8	67.3	77.1
EfficientLPS[31]	59.2	65.1	69.8	75.0	58.0	68.2	78.0	**60.9**	**71.0**	72.8
MaskPLS-M [24]	59.8	-	69.0	76.3	-	-	-	-	-	-
CenterLPS (Ours)	**62.1**	**67.0**	**72.0**	**80.7**	68.4	**75.2**	**91.0**	57.5	69.7	73.2

Table 2: LiDAR panoptic segmentation results on the test set of SemanticKITTI. † denotes the results with TTA and model ensemble. All results in [%].

Method	PQ	PQ†	RQ	SQ	PQ$^{\mathrm{Th}}$	RQ$^{\mathrm{Th}}$	SQ$^{\mathrm{Th}}$	PQ$^{\mathrm{St}}$	RQ$^{\mathrm{St}}$	SQ$^{\mathrm{St}}$
RangeNet++ [28] + PointPillars [17]	37.1	45.9	47.0	75.9	20.2	25.2	75.2	49.3	62.8	76.5
LPSAD [27]	38.0	47.0	48.2	76.5	25.6	31.8	76.8	47.1	60.1	76.2
KPConv [34] + PointPillars [17]	44.5	52.5	54.4	80.0	32.7	38.7	81.5	53.1	65.9	79.0
Panoster [9]	52.7	59.9	64.1	80.7	49.4	58.5	83.3	55.1	68.2	78.8
Panoptic-PolarNet [46]	54.1	60.7	65.0	81.4	53.3	60.6	87.2	54.8	68.1	77.2
CPSeg [18]	57.0	63.5	68.8	82.2	55.1	64.1	86.1	58.4	72.3	79.3
DS-Net [14]	55.9	62.5	66.7	82.3	55.1	62.8	87.2	56.5	69.5	78.7
EfficientLPS [31]	57.4	63.2	68.7	83.0	53.1	60.5	87.8	60.5	74.6	79.5
SCAN [38]	61.5	67.5	72.1	84.5	61.4	69.3	88.1	61.5	74.1	81.8
Panoptic-PHNet [20]	61.5	67.9	72.1	84.8	63.8	70.4	**90.7**	59.5	73.3	80.5
MaskPLS-M [24]	58.2	69.3	68.6	83.9	55.7	61.7	89.2	60.0	73.7	80.0
CenterLPS (Ours)	61.6	67.9	72.6	84.0	63.8	71.8	88.4	60.0	73.2	80.8
CenterLPS† (Ours)	**65.4**	**71.4**	**76.0**	**85.3**	**68.0**	**75.8**	89.5	**63.4**	**76.2**	**82.2**

Table 3: LiDAR panoptic segmentation results on the validation set of nuScenes. All results in [%].

Method	PQ	PQ†	RQ	SQ	PQ$^{\mathrm{Th}}$	RQ$^{\mathrm{Th}}$	SQ$^{\mathrm{Th}}$	PQ$^{\mathrm{St}}$	RQ$^{\mathrm{St}}$	SQ$^{\mathrm{St}}$
PanopticTrackNet [15]	51.4	56.2	63.3	80.2	45.8	55.9	81.4	60.4	75.5	78.3
DS-Net [14]	42.5	51.0	50.3	83.6	32.5	38.3	83.1	59.2	70.3	84.4
EfficientLPS [31]	62.0	65.6	73.9	83.4	56.8	68.0	83.2	70.6	83.6	83.8
Panoptic-PolarNet [46]	63.4	67.2	75.3	83.9	59.2	70.3	84.1	70.4	83.5	83.6
GP-S3Net [30]	61.0	67.5	72.0	84.1	56.0	65.2	85.3	66.0	78.7	82.9
PVCL [23]	64.9	67.8	77.9	81.6	59.2	72.5	79.7	67.6	79.1	77.3
SCAN [38]	65.1	68.9	75.3	85.7	60.6	70.2	85.7	72.5	83.8	85.7
Panoptic-PHNet [20]	74.7	77.7	84.2	**88.2**	74.0	82.5	**89.0**	**75.9**	86.9	**86.8**
MaskPLS-M [24]	57.7	60.2	66.0	71.8	64.4	73.3	84.8	52.2	60.7	62.4
CenterLPS (Ours)	**76.4**	**79.2**	**88.0**	86.2	**77.5**	**88.4**	87.1	74.6	**87.3**	84.9

characteristics of instances with the center feature embedding and decodes instance masks with dynamic convolution.

4.3 Ablation Study

Baseline. We build a strong baseline by taking the dynamic shift (DS) module [14] to perform instance segmentation. For a fair comparison, the baseline keeps the same semantic prediction and offset regression as our CenterLPS. As presented in Table 4, the baseline achieves 62.0% in terms of PQ$^{\mathrm{Th}}$.

Effectiveness of proposed components. We analyze the effect of the proposed sparse center proposal network (SCPN), mask fusion module (MF), center-aware transformer (CTR), and dynamic convolution. The results are presented in Table 4 and show that our model (Variant 1) equipped with SCPN and the vanilla dynamic convolution (without any priors such as PP and SP) has already

Table 4: Effect of the components.

Variants	SCPN	MF	CTR	PP	SP	PQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU
baseline						59.4	62.0	69.7	87.4	66.2
1	✓					59.3	62.5	70.9	85.4	68.1
2	✓	✓				60.4	65.1	73.1	87.9	67.9
3	✓	✓	✓			61.0	66.4	73.4	89.4	67.9
4	✓	✓	✓	✓		61.7	67.5	75.1	88.4	68.1
5	✓	✓	✓	✓	✓	62.1	68.4	75.2	91.0	68.1

Table 5: Ablation study on different backbones.

Dataset	Backbone	PQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU
Sem.KITTI	GASN	62.1	68.4	75.2	91.0	68.1
	SPVCNN	60.7	66.4	74.1	90.0	68.0
nuScenes	GASN	76.4	77.5	88.4	87.1	77.1
	SPVCNN	75.7	77.2	88.1	87.0	76.7

achieved high performance and surpasses the baseline by 0.5% and 1.9% in terms of PQ^{Th} and $mIoU$, respectively. It demonstrates the effectiveness of our paradigm of center-based instance encoding and decoding. When using the mask fusion module (MF) to process the predictions, the performance can be boosted by 2.6% on PQ^{Th} (Variant 2 vs. Variant 1), showing that our MF module can effectively improve the panoptic quality. Furthermore, we demonstrate the effectiveness of the center-aware transformer (CTR). We achieve the 1.3% gain on PQ^{Th} when using CTR to collect the context between different center feature embedding and around centers (Variant 3 vs. Variant 2). We also provide the results that use position prior/embedding (PP) for dynamic convolution as [13]. As shown in Table 4 (Variant 4), PP improves the performance by 1.1% in terms of PQ^{Th}. Finally, we analyze the effect of the proposed shape prior (SP) to dynamic convolution. Comparing Variant 5 with Variant 4 in Table 4, we can see that SP boosts the performance by 0.9% on PQ^{Th}, demonstrating that shape prior can further enhance the mask features and expedite the convergence of the network.

Effect of different backbones. The backbone extracts point-wise features for semantic prediction and instance segmentation. We provide detailed experiments in Table 5 to show the effect of different backbones, *i.e.*, grid-based GASN [41] and hybrid-based SPVCNN [33] using sparse point-voxel convolutions. Both backbones have 4 scales and 64 dimensions for SemanticKITTI, and 6 scales and 128 dimensions for nuScenes. Our models with different backbones achieve comparable performance on both SemanticKITTI and nuScenes validation, demonstrating that our approach can be applied to other backbones. And we experimentally find that GASN performs better in our center-based framework for LPS. We explain that deep supervision and multi-scale geometry feature enhance the representation ability of GASN, especially on small objects, which is vital for downstream tasks such as instance segmentation.

Prior analysis of Dynamic Convolution. Positional information is important for distinguishing between different instances, and the coarse shape of instances can effectively guide the dynamic convolution network focus on the objects. Thus, we concatenate the offsets from points to the instance centers and the generated coarse mask with point features to inject the position and shape priors.

Table 6: Effectiveness of priors for dynamic convolution.

PP	SP	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU
×	×	61.0	71.1	80.0	66.4	73.4	89.4	67.9
✓	×	61.7	72.0	79.6	67.5	75.1	88.4	68.1
×	✓	62.0	72.0	80.6	68.3	75.2	90.8	67.9
✓	✓	62.1	72.0	80.7	68.4	75.2	91.0	68.1

Table 7: Kernel shape analysis on dynamic convolution.

kernel shape	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU
[1]	61.9	71.9	80.6	67.9	74.9	90.7	68.0
[16, 1]	62.1	72.0	80.7	68.4	75.2	91.0	68.1
[16, 8, 1]	62.1	72.1	80.4	68.4	75.4	90.4	68.2
[16, 16, 1]	62.0	72.1	80.5	68.3	75.3	90.5	68.2

We provide further analysis of the priors in dynamic convolution in Table 6. Without both position and shape priors, the model's performance drops from 62.1% to 61.0% in terms of PQ. And the position prior contributes 0.7% and 1.1% gains on PQ and PQ^{Th}, respectively (line 2 vs. line 1). While the mask prior boosts the performance by 1.0% on PQ and 1.9% on PQ^{Th} (line 3 vs. line 1). The results show that both position and shape priors are beneficial, and the latter plays a more important role in improving the segmentation quality. We explain that the shape prior implies more abundant information, including position and geometric cues of instances.

Kernel shape analysis. The dynamic convolution generates kernel weights based on the enhanced center feature embedding. The kernel weights are further used to initialize the weights of the shallow convolution network $\mathcal{F}_w$ for decoding the final masks from the mask features F_m. To analyze the effectiveness of kernel shape, we change the output channels and the number of layers of convolution network $\mathcal{F}_w$. The results are presented in Table 7 and show that CenterLPS is robust to the kernel shape.

5 CONCLUSION

This paper focuses on LiDAR Panoptic Segmentation (LPS). Unlike mainstream detection- and clustering-based methods, we propose a new clustering-free and detection-free framework, dubbed CenterLPS, with the center-based instance encoding and decoding paradigm. Specifically, a sparse center proposal network is devised to generate the sparse 3D instance center and feature embedding to encode the characteristics of instances. Then a center-aware transformer is applied to collect the context between the center feature embedding and around centers. Moreover, we utilize dynamic convolution to generate kernel weights and decode the final instance masks. A mask fusion module is also devised to unify the semantic and instance predictions and improve the panoptic quality. Extensive experiments on multiple benchmarks demonstrate the effectiveness of our CenterLPS.

6 ACKNOWLEDGMENTS

This work was supported by a Grant from The National Natural Science Foundation of China (No. U21A20484).

REFERENCES

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. 2019. SemanticKITTI: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE Int'l Conf. on Computer Vision*. 9297–9307.

[2] Jens Behley, Andres Milioto, and Cyrill Stachniss. 2021. A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI. In *2021 IEEE Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 13596–13603.

[3] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*. 4413–4421.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conf., Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 213–229.

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 1290–1299.

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. 2021. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems* 34 (2021), 17864–17875.

[7] Fabian Duerr, Hendrik Weigel, and Jürgen Beyerer. 2022. RangeBird: Multi View Panoptic Segmentation of 3D Point Clouds with Neighborhood Attention. In *2022 Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 11131–11137.

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

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

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conf. on computer vision and pattern recognition*. IEEE, 3354–3361.

[11] Yi Gu, Yuming Huang, Chengzhong Xu, and Hui Kong. 2022. MaskRange: A Mask-classification Model for Range-view based LiDAR Segmentation. *arXiv preprint arXiv:2206.12073* (2022).

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE Int'l Conf. on computer vision*. 2961–2969.

[13] Tong He, Chunhua Shen, and Anton Van Den Hengel. 2021. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *Proceedings of the IEEE/CVF Conf. on computer vision and pattern recognition*. 354–363.

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

[15] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. 2020. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189* (2020).

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conf. on computer vision and pattern recognition*. 12697–12705.

[18] Enxu Li, Ryan Razani, Yixuan Xu, and Bingbing Liu. 2021. CPSeg: Cluster-free Panoptic Segmentation of 3D LiDAR Point Clouds. *arXiv preprint arXiv:2111.01723* (2021).

[19] Enxu Li, Ryan Razani, Yixuan Xu, and Bingbing Liu. 2022. Smac-seg: Lidar panoptic segmentation via sparse multi-directional attention clustering. In *2022 Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 9207–9213.

[20] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. 2022. Panoptic-PHNet: Towards Real-Time and High-Precision LiDAR Panoptic Segmentation via Clustering Pseudo Heatmap. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 11809–11818.

[21] Xiaoyan Li, Gang Zhang, Boyue Wang, Yongli Hu, and Baocai Yin. 2023. Center Focusing Network for Real-Time LiDAR Panoptic Segmentation. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 13425–13434.

[22] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. 2021. Fully convolutional networks for panoptic segmentation. In *Proceedings of the IEEE/CVF Conf. on computer vision and pattern recognition*. 214–223.

[23] Minzhe Liu, Qiang Zhou, Hengshuang Zhao, Jianing Li, Yuan Du, Kurt Keutzer, Li Du, and Shanghang Zhang. 2022. Prototype-Voxel Contrastive Learning for LiDAR Point Cloud Panoptic Segmentation. In *2022 Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 9243–9250.

[24] Rodrigo Marcuzzi, Lucas Nunes, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. 2023. Mask-Based Panoptic LiDAR Segmentation for Autonomous Driving. *IEEE Robotics and Automation Letters* (2023).

[25] Jianbiao Mei, Mengmeng Wang, Yeneng Lin, Yi Yuan, and Yong Liu. 2021. Transvos: Video object segmentation with transformers. *arXiv preprint arXiv:2106.00588* (2021).

[26] Jianbiao Mei, Yu Yang, Mengmeng Wang, Xiaojun Hou, Laijian Li, and Yong Liu. 2023. PANet: LiDAR Panoptic Segmentation with Sparse Instance Proposal and Aggregation. *arXiv preprint arXiv:2306.15348* (2023).

[27] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. 2020. Lidar panoptic segmentation for autonomous driving. In *2020 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 8505–8512.

[28] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. 2019. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ Int'l Conf. on intelligent robots and systems (IROS)*. IEEE, 4213–4220.

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

[30] Ryan Razani, Ran Cheng, Enxu Li, Ehsan Taghavi, Yuan Ren, and Liu Bingbing. 2021. GP-S3Net: Graph-based panoptic sparse semantic segmentation network. In *Proceedings of the IEEE/CVF Int'l Conf. on Computer Vision*. 16076–16085.

[31] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. 2021. Efficientlps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics* 38, 3 (2021), 1894–1914.

[32] Shihao Su, Jianyun Xu, Huanyu Wang, Zhenwei Miao, Xin Zhan, Dayang Hao, and Xi Li. 2023. PUPS: Point Cloud Unified Panoptic Segmentation. *arXiv preprint arXiv:2302.06185* (2023).

[33] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. 2020. Searching efficient 3d architectures with sparse point-voxel convolution. In *Computer Vision–ECCV 2020: 16th European Conf., Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII*. Springer, 685–702.

[34] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF Int'l Conf. on computer vision*. 6411–6420.

[35] Zhi Tian, Chunhua Shen, and Hao Chen. 2020. Conditional convolutions for instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conf., Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 282–298.

[36] Yizheng Wu, Min Shi, Shuaiyuan Du, Hao Lu, Zhiguo Cao, and Weicai Zhong. 2022. 3D Instances as 1D Kernels. In *Computer Vision–ECCV 2022: 17th European Conf., Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*. Springer, 235–252.

[37] Zeqi Xiao, Wenwei Zhang, Tai Wang, Chen Change Loy, Dahua Lin, and Jiangmiao Pang. 2023. Position-Guided Point Cloud Panoptic Segmentation Transformer. *arXiv preprint arXiv:2303.13509* (2023).

[38] Shuangjie Xu, Rui Wan, Maosheng Ye, Xiaoyi Zou, and Tongyi Cao. 2022. Sparse cross-scale attention network for efficient lidar panoptic segmentation. In *Proceedings of the AAAI Conf. on Artificial Intelligence*, Vol. 36. 2920–2928.

[39] Yixuan Xu, Hamidreza Fazlali, Yuan Ren, and Bingbing Liu. 2023. AOP-Net: All-in-One Perception Network for Joint LiDAR-based 3D Object Detection and Panoptic Segmentation. *arXiv preprint arXiv:2302.00885* (2023).

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

[41] Maosheng Ye, Rui Wan, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. 2022. Efficient Point Cloud Segmentation with Geometry-Aware Sparse Networks. In *Computer Vision–ECCV 2022: 17th European Conf., Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, 196–212.

[42] Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. 2021. Drinet: A dual-representation iterative learning network for point cloud segmentation. In *Proceedings of the IEEE/CVF Int'l Conf. on computer vision*. 7447–7456.

[43] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. 2021. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conf. on computer vision and pattern recognition*. 11784–11793.

[44] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. 2021. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems* 34 (2021), 10326–10338.

[45] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as points. *arXiv preprint arXiv:1904.07850* (2019).

[46] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. 2021. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 13194–13203.

[47] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. 2022. Centerformer: Center-based transformer for 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conf., Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*. Springer, 496–513.

Algorithm 1: Algorithm for the process of mask fusion

```
Input: Mask decoding: $\{M_{c_1}, ..., M_{c_{N_c}}\} \in \mathbb{R}^{N_c \times N}$; Instance
       categories: $\{s_{c_1}, ..., s_{c_{N_c}}\} \in \mathbb{R}^{N_c}$; Semantic
       predictions: $M_{sem} \in \mathbb{R}^{N}$.
Output: Unified panoptic results: $\{M_{sem}, M_{id}\}$.
Calculate the confidence score $\{r_1, ..., r_{N_c}\}$ for each mask:
  $r_i = \text{Avg}(\mathbf{1}_{\{M_{c_i} > 0.5\}} M_{c_i})$ for $i \in [1, N_c]$
Binarize the mask decoding by: $M_{c_i} = M_{c_i} > 0.5$ for
  $i \in [1, N_c]$
Calculate the overlap score matrix $H$: $H_{ij} = IoU(M_{c_i}, M_{c_j})$
  for $i \in [1, N_c]$ and $j \in [1, N_c]$;
Construct the connectivity matrix $O$: $O_{ij} = (H_{ij} > \theta_{thres})$
  and $(s_{c_i} == s_{c_j})$ for $i \in [1, N_c]$ and $j \in [1, N_c]$;
Find the $N_g$ groups $\{G_1, .., G_{N_g}\}$ by connected-component
  labeling algorithm according to $O$; Each group contains
  masks with overlap scores greater than $\theta_{thres}$ and the
  same semantic categories;
Obtain the merged mask $\{M_1, ..., M_{N_g}\}$ with categories
  $\{s_1, ..., s_{N_g}\}$ by: $M_i = \cup G_i$ for $i \in [1, N_g]$;
update the confidence scores $\{r_1, ..., r_{N_g}\}$ by averaging the
  scores of masks in the same group;
Sort the merged mask based on confidence score
  $\{r_1, ..., r_{N_g}\}$;
Initialize instance IDs: $M_{id} = zeros(N)$; $id = 1$;
for $k \leftarrow 1$ to $N_g$ do
    if sum$(M_k) < N_{keep}$ then
        continue
    $M_{sem}[M_k] = s_k$
    $M_{id}[M_k] = id$
    $id$ += 1
end
return $M_{sem}$, $M_{id}$
```

A DATASETS AND METRICS

SemanticKITTI. SemanticKITTI [1] is derived from the KITTI [10] odometry dataset and includes 22 LiDAR sequences (10, 1, and 10 for training, validation, and testing, respectively) captured by a 64-beam LiDAR sensor. It provides point-wise labels for LiDAR-based panoptic segmentation. And there are 19 annotated classes, including 8 *thing* classes and 11 *stuff* classes.

nuScenes. nuScenes [8] is a large-scale urban driving dataset, which includes 1000 LiDAR scenes of 20s duration captured by a 32-beam LiDAR sensor. The dataset consists of 850 scenes for training and validation, as well as 150 scenes for testing. For LiDAR-based panoptic segmentation task, it involves 16 annotated point-wise labels, including 10 *thing* categories and 6 *stuff* categories.

Metrics. The metrics [2] for LPS include Panoptic Quality (PQ), Segmentation Quality (SQ), and Recognition Quality (RQ). We also calculate these metrics independently for *thing* and *stuff* classes. For *thing* class, the metrics are denoted by PQ^{Th}, SQ^{Th}, RQ^{Th}, and for *stuff* class, they are denoted by PQ^{St}, SQ^{St}, RQ^{St}. Also, we report $\text{PQ}^{\dagger}$ defined in [29] by swapping the PQ of each stuff class to *IoU* and then averaging over all classes for *stuff* classes. The Mean *IoU* (*mIoU*) evaluates the quality of semantic segmentation.

Table 8: Comparison between NMS with pasting strategy (NMS-P) and our mask fusion module (MF).

variants	PQ	$\text{PQ}^{\dagger}$	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU	latency (ms)
NMS	59.9	65.1	63.9	72.0	86.0	**68.1**	**11.6**
NMS-P	60.6	65.8	65.4	72.7	88.7	67.9	15.6
MF	**61.0**	**66.2**	**66.4**	**73.4**	**89.4**	67.9	15.1

Table 9: Architecture analysis of transformer on SemanticKITTI validation.

PE	SA	LCA	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU
×	✓	✓	60.7	70.9	79.8	65.9	73.3	89.0	67.8
✓	×	✓	60.5	70.7	79.9	65.4	72.8	89.1	67.8
✓	✓	×	60.6	70.8	79.9	65.6	73.0	89.1	67.9
✓	✓	✓	**61.0**	**71.1**	**80.0**	**66.4**	**73.4**	**89.4**	**67.9**

Table 10: Comparison of different blocks and head configurations of the transformer.

block	head	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU
1	8	60.9	70.9	80.1	66.5	73.6	89.6	67.7
2	4	61.0	71.1	80.0	66.4	73.4	89.4	67.9
2	8	60.9	70.9	80.1	66.3	73.4	89.6	67.7
3	4	60.8	70.9	80.1	66.2	73.4	89.5	67.9
3	8	60.7	70.7	80.0	65.9	72.9	89.4	67.6
4	4	60.9	70.9	80.1	66.3	73.4	89.6	67.6

B ANALYSIS

We conduct a series of experiments to provide further analysis on mask fusion and the center-aware transformer. All variants are equipped with vanilla dynamic convolution (without position/shape priors) for a fair comparison. The efficiency analysis is also provided.

B.1 Analysis of Mask Fusion Module

The detailed procedure of our mask fusion is presented in Algorithm 1. The merging process is performed through the connected-component labeling algorithm, which can be implemented by the efficient depth-first algorithm. The merged masks are pasted on the semantic predictions by order of confidence score. And masks that contain points less than N_{keep} are dropped. We also provide a further comparison between NMS incorporating our pasting strategy (NMS-P) and our mask fusion (MF) in Table 8. The results demonstrate the effectiveness of our mask fusion module, including the merging and pasting strategies.

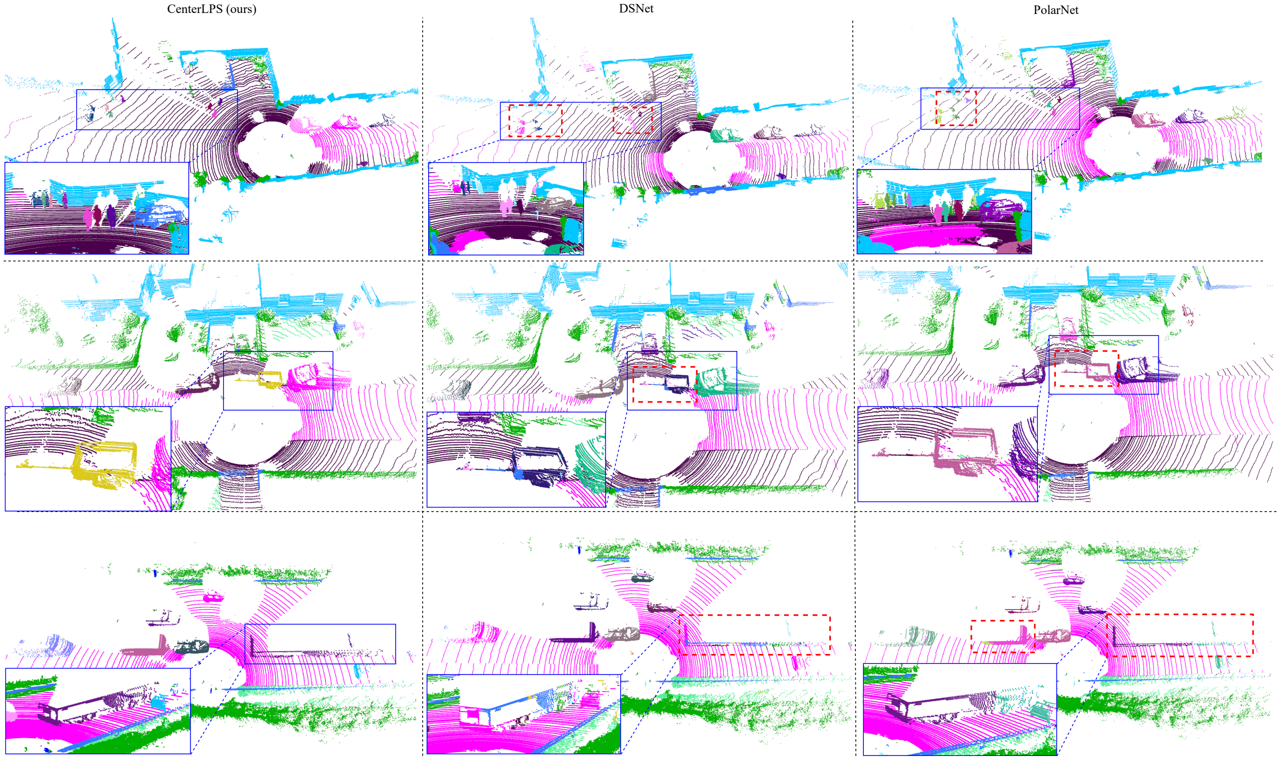

Figure 5: Visualizations on the SemanticKITTI test set. Our CenterLPS handles close instances and large objects better.

Table 11: Ablation on k in k-nearest neighbors.

k	PQ	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	mIoU
32	60.6	70.8	79.8	65.6	73.1	88.9	67.6
64	61.0	71.1	80.0	66.4	73.4	89.4	67.9
96	60.9	71.1	79.8	66.2	73.8	88.9	67.8
128	60.9	71.0	80.0	66.3	73.7	89.2	67.9

B.2 Ablation on Center-aware Transformer

Effect of different components. Position embedding, self-attention, and local cross-attention are important components of the transformer. Position embedding provides the spatial relationships between different attentive points. Self-attention models the inter-instance dependencies, and local cross-attention collects the context around the centers. We explore the effect of these components. The results are presented in Table 9 and show that all these components contribute to the performance of the center-aware transformer. And self-attention plays a more important role, meaning inter-instance relationships help distinguish different instances better.

Ablation on encoder blocks and attention heads. Table 10 presents the detailed results of using a different number of encoder blocks and attention heads for the center-aware transformer. The results show that more transformer encoder blocks and attention heads do not assure better performance. The transformer model with 2 blocks and 4 heads performs best.

Effect of different k in k-nearest neighbors. In the local cross-attention layer of the center-aware transformer, k-nearest neighbors are used to collect context around centers for an intra-instance relationship. On the other way, using k-nearest neighbors also reduces computation consumption. We change the k to investigate the effect. As shown in Table 11, it performs best with $k = 64$.

B.3 Efficiency Analysis

We perform runtime experiments on a single NVIDIA 1080 TI GPU. The mean value over the SemanticKITTI validation set is reported. The backbone has a runtime of 84.2 ms, and the center-based instance segmentation and mask fusion add 70.1 ms and 14.2 ms, respectively. Specifically, the sparse center proposal network, center-aware transformer, and dynamic convolution require 30.2 ms, 6.2 ms, and 33.7 ms, respectively. Compared to the two representative methods, *i.e.*, clustering-based DSNet [14] and detection-based EfficientLPS [31], which runs 474.5 ms and 212.8 ms in the inference stage, our CenterLPS has lower latency. We test DSNet with the same backbone as our CenterLPS on the same platform.

C QUALITATIVE RESULTS

We show visual comparisons of our CenterLPS with DSNet [14] and Panoptic-PolarNet [46] on the SemanticKITTI test set. These examples show that our approach performs well not only for crowded scenes but also for big objects. Specifically, the adjacent objects (such as people) in the upper part of Fig.5 are accurately distinguished. And the large objects (such as buses) in the bottom part of Fig.5 are correctly segmented without the over-segment problem.