Panoptic-PHNet: Towards Real-Time and High-Precision LiDAR Panoptic Segmentation via Clustering Pseudo Heatmap

Jinke Li Xiao He Yang Wen Yuan Gao Xiaoqiang Cheng Dan Zhang
Uisee Foundation Research & Development
{jinke.li, xiao.he, yang.wen, yuan.gao, xiaoqiang.cheng, dan.zhang}@uisee.com

Abstract

*As a rising task, panoptic segmentation is faced with challenges in both semantic segmentation and instance segmentation. However, in terms of speed and accuracy, existing LiDAR methods in the field are still limited. In this paper, we propose a fast and high-performance LiDAR-based framework, referred to as Panoptic-PHNet, with three attractive aspects: 1) We introduce a clustering pseudo heatmap as a new paradigm, which, followed by a center grouping module, yields instance centers for efficient clustering without object-level learning tasks. 2) A knn-transformer module is proposed to model the interaction among foreground points for accurate offset regression. 3) For backbone design, we fuse the fine-grained voxel features and the 2D Bird's Eye View (BEV) features with different receptive fields to utilize both detailed and global information. Extensive experiments on both SemanticKITTI dataset and nuScenes dataset show that our Panoptic-PHNet surpasses state-of-the-art methods by remarkable margins with a real-time speed. We achieve the **1st place** on the public leaderboard of SemanticKITTI and leading performance on the recently released leaderboard of nuScenes.*

1. Introduction

In recent years, there has been a rapid development of autonomous driving, and as an essential perception task of its key technologies, scene understanding has attracted considerable attention from researchers. Panoptic segmentation is a recently introduced task in the image domain [18] that aims to unify semantic segmentation and instance segmentation in a single framework. With the release of LiDAR point cloud benchmarks, e.g., SemanticKITTI [1] and nuScenes [10], related works in the 3D field have also been extensively promoted.

The purpose of LiDAR panoptic segmentation is not only to predict class labels for all points, including foreground points (*thing*) and background points (*stuff*), but also the instance IDs for *thing* points. According to the implementation of instance segmentation, panoptic segmentation can be categorized into proposal-based and proposal-free approaches.

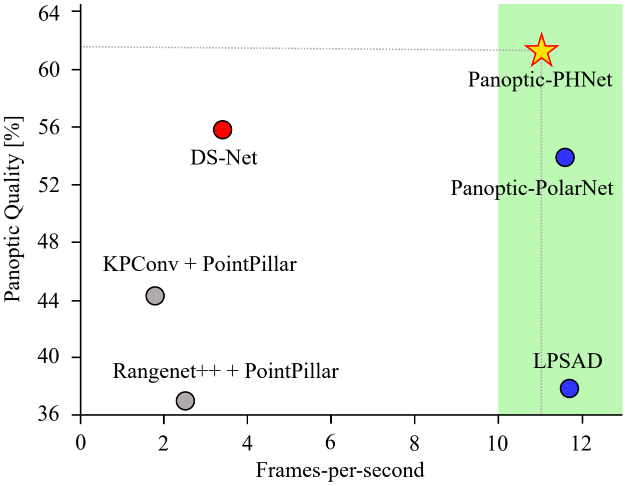


Figure 1. Panoptic quality vs. single frame inference latency on SemanticKITTI [1]. Green area indicates real-time zone, which meets 10 frame-per-second frequency. The 2D CNN based approaches [22, 40], the 3D CNN based approach [14] and the combined methods [1] are shown in blue, red and gray respectively. Our proposed Panoptic-PHNet outperforms all other methods in PQ by a large margin and still maintains a real-time speed.

Proposal-based methods require an independent network or branch to predict proposals [1, 17], a drawback of which is that the capability of instance segmentation heavily depends on the performance of object detection.

In contrast, proposal-free ones [14, 19, 24] explore clustering-based methods for instance segmentation. Since such methods are not bothered with the inconsistency issue based on a cascade design, they are relatively elegant in respect of implementation. Yet the commonly used heuristic clustering algorithms, such as Mean Shift [7] and HDBSCAN [4], are time-consuming and difficult to be accelerated with GPU. Although Panoptic-PolarNet [40] attempts to use a lightweight instance head from Panoptic-DeepLab

[5] to predict a center heatmap and offsets, the centers predicted by the heatmap branch may not match the clustered locations from the offset branch. The possible inconsistency between these two independent branches limits such a method.

In general, the proposal-free methods learn the offset for each *thing* point, with which the point can be shifted to be near its instance center. Projecting all the shifted points onto a BEV pseudo image, we find that it shares a similar pattern with a learned center heatmap as Panoptic-PolarNet [40]. In other words, the projected image can be a natural heatmap to determine the existence of instances.

Specifically, given shifted *thing* points, we propose to create a *clustering pseudo heatmap* by projecting these points onto a BEV image directly, the number of points in each pixel, which we call the quantitative density, represents the corresponding score. Thus, through the proposed pseudo heatmap, instance centers can be easily yielded by a window-based max-pooling and used to cluster all the *thing* points as [5]. In this way, we remove the separate heatmap learning branch and the inconsistency issue mentioned above are hence eliminated. What you see is what you get, an object-level center is generated as long as there is a cluster of points.

Nevertheless, there are also such cases where multiple centers belonging to one instance may be generated due to inaccurate point offset regression. We further propose a *center grouping module*, which integrates such redundant centers, to maintains the completeness of instances.

Moreover, it is obvious that high-quality offset regression leads to better shifted points for our clustering pseudo heatmap. Thus, inspired by Transformer [31], which is popular in the natural language processing and computer vision fields, we introduce a *knn-transformer module* to model the interaction of *thing* points efficiently. Focusing on the relationship of spatial distance among local points, our knn-transformer promotes the offset regression with low computational consumption.

Regarding the design of backbone, we aggregate features at different scales more flexibly considering both accuracy and inference speed. We first extract fine-grained voxel features, which are then encoded in 2D BEV space with different receptive fields via a Unet-like network as PolarNet [37] for real-time purpose. The 2D BEV features are further mapped back to each voxel with the height dimension. By concatenating, the obtained voxel-wise features contain not only the 2D encoded features at different BEV scales, but also the fine-grained voxel features.

We evaluate our Panoptic-PHNet on SemanticKITTI and nuScenes datasets. Extensive experiments show that our approach outperforms all the state-of-the-art methods on both of the two benchmarks (1st place on the public leaderboard of SemanticKITTI) with a real-time latency.

Our contributions are summarized as below:

- We propose a clustering pseudo heatmap generated from the shifted *thing* points directly without extra learning tasks, which allows us to avoid the inconsistency issue between two independent branches and accelerate the clustering process. A center grouping module is further introduced to maintain the integrity of instances.
- We propose a knn-transformer module to efficiently model the interaction among *thing* points for accurate offset regression.
- We present a backbone network fusing the fine-grained voxel features and the BEV features at different scales, which, compared with the networks utilizing pure BEV features, improves the final accuracy significantly at the cost of only a little time consumption.
- Experiments show that our approach achieves state-of-the-art performance on both SemanticKITTI and nuScenes datasets in real time, as shown in Fig. 1.

2. Related Work

2.1. Representation Learning of Point Cloud

As effective data representation is the foundation of learning based tasks, for irregular and sparse point clouds, there are generally two ways to learn representations in previous studies. One is to learn features directly at the point level, while the other regularizes raw point clouds at first before feature extraction. Based on PointNet [26] and PointNet++ [27], KPConv [30] and RandLA [16] process the irregular point clouds straightforwardly, which, however, takes a time-consuming preprocess to build the graph among points. VoxelNet [39] first projects point clouds to regular voxels and utilizes 3D CNNs to learn features. SECOND [35] introduces sparse convolution to promote learning efficiency for voxel-wise features. To further optimize the latency of feature extraction as well as the memory consumption, PointPillars [20] collapses the height dimension via PointNet and then treats the input as a BEV image. PolarNet [37] takes the imbalanced distribution of points in physical space into consideration and encodes point clouds into a polar BEV map. Range image [9, 23, 33, 34] is another common projection space for efficient feature encoding, yet the 3D topological relations are also weakened. Methods [29, 32, 38] propose to fuse information from different perspectives.

2.2. LiDAR Panoptic Segmentation

As a newly proposed research domain, panoptic segmentation unifies semantic segmentation and instance segmen-

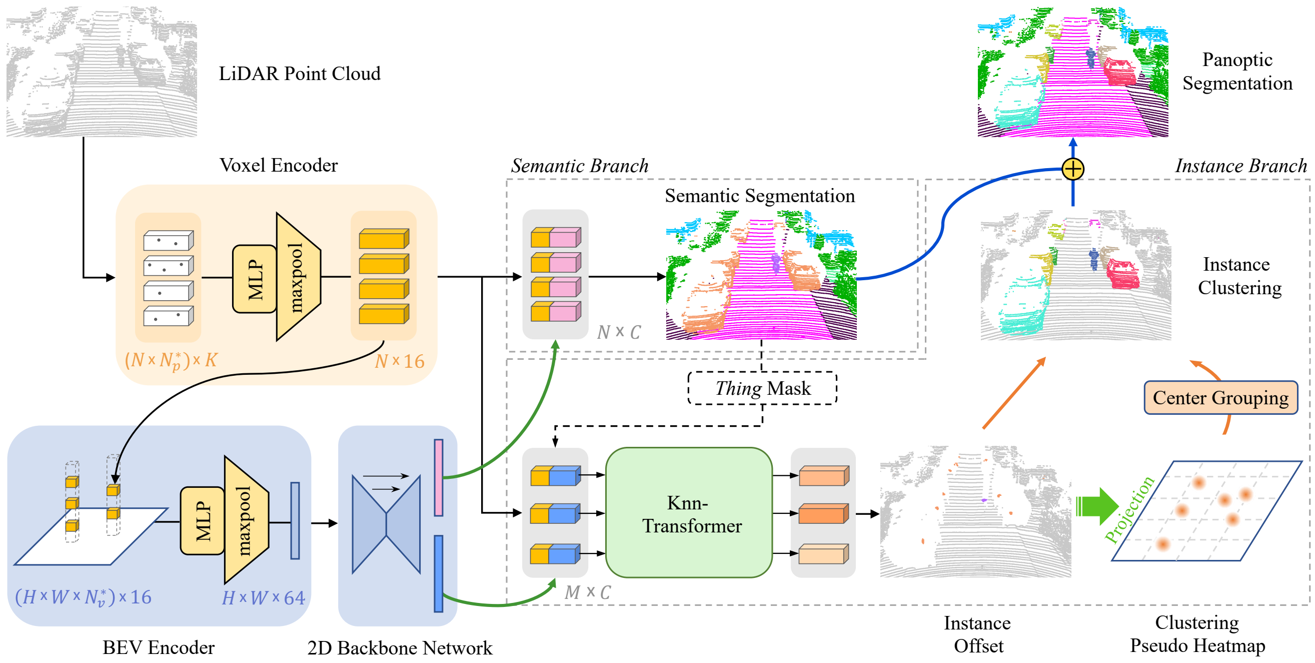


Figure 2. The overall framework of our Panoptic-PHNet. The backbone consists of a voxel encoder, a BEV encoder and a 2D backbone network for feature extraction. The extracted BEV features are concatenated with the fine-grained voxel features as voxel representations for semantic and instance branches. In the instance branch, a knn-transformer module is introduced to model the interaction among *thing* voxels. A clustering pseudo heatmap is generated from the shifted *thing* voxels to yield instance centers followed by a center grouping module. Finally, the outputs of the two branches are combined via a voting-based scheme to obtain the panoptic segmentation results.

tation. In terms of the approaches of processing ID information, two frameworks, i.e., proposal-based and proposal-free, are designed.

Proposal-based panoptic segmentation. PanopticTrackNet [17] utilizes Mask R-CNN [13] for instance segmentation, and attaches a semantic head to classify the *stuff* points. SemanticKITTI [1] and nuScenes [10] release LiDAR panoptic segmentation datasets and report results by jointing existing state-of-the-art object detectors and semantic segmentation networks. For proposal-based methods, although the predicted bounding boxes make it easy to segment instances, the final performance depends heavily on the object detection task.

Proposal-free panoptic segmentation. Panoptic-PolarNet [40] adopts a lightweight instance head from Panoptic-DeepLab [5] to predict instance centers and point offsets without bounding box regression. However, it is still limited due to the inconsistency issue between two independent branches and the possible failures of instance center prediction. There are also studies [11, 14, 22] using pure clustering for instance segmentation. DS-Net [14] proposes a dynamic shifting module to shift points towards the instance centers in an iterative manner and utilizes the Mean Shift clustering to segment instances. It should be noted that taken as post process, conventional clustering methods are usually time-consuming.

3. Methodology

3.1. Overview

As pixels are the essential elements for Unet, voxels are the basic units in our network. Following the design of Unet, which fuses low-level and high-level features for each pixel, our network aggregates both 2D semantic features under different receptive fields and fine-grained 3D features for each voxel. The former quickens the convergence of the task and the latter facilitates distinguishing different voxels from each other.

The framework of our Panoptic-PHNet is shown in Fig. 2. The input LiDAR point cloud is first encoded through a voxel encoder to be 3D voxel representations, which are further transformed into size-fixed 2D representations by a BEV encoder. A Unet-like 2D backbone network is utilized to extract BEV features with different receptive fields. The BEV features are gathered for each voxel according to their coordinates. New features are generated by concatenating the gathered BEV features with the low-level fine-grained voxel features, which are then fed into two branches for semantic and instance segmentation respectively. In the instance branch, a knn-transformer module is introduced for modeling the interaction among *thing* voxels to enhance the feature representation. We predict offsets towards instance centers to shift *thing* voxels, subsequently a clustering pseudo heatmap is generated by projection according to the quantitative density of shifted voxels to yield instance centers. The possible redundant centers are integrated through a center grouping module. At last, combining the outputs of two branches, we obtain the final panoptic segmentation results via a voting-based scheme as [40]. In the following sections, we first elaborate on two components in the instance branch of our Panoptic-PHNet, then the backbone design is presented.

3.2. Clustering Pseudo Heatmap

After the center offsets prediction and the voxels shifting, the remaining work in the instance branch can be regarded as a clustering task. For efficient instance clustering, we proceed in the BEV space based on the assumption as [40] that the interested *thing* objects are separated from each other and do not overlap under the bird's eye view.

To address the current issues of existing methods as analyzed in Sec. 1, we propose a clustering pseudo heatmap to yield instance centers by projecting the shifted *thing* voxels onto a BEV map without extra learning branches. Since the number of voxels is used as the score for each BEV grid, the location with the most voxels in a local area corresponds to a local peak on the pseudo heatmap, which can be naturally taken as an instance center. Such a bottom-up design ensures the consistency between the instance centers and the shifted voxels. More specifically, we project the shifted *thing* voxels $V' \in \mathbb{R}^{M\times 3}$ to a BEV pseudo image $I' \in \mathbb{R}^{H\times W\times C_n}$ based on their locations on the BEV map, where M is the number of *thing* voxels, H and W are the size of the BEV map and C_n is the number of the semantic categories. By summing the voxel number along the dimension of C_n, a class-agnostic pseudo heatmap $I \in \mathbb{R}^{H\times W\times 1}$ is created. As a non-maximum suppression, a window-based 2D max pooling is adopted to efficiently pick out the local centers. Compared with the dense learning-based heatmap in [40], our clustering pseudo heatmap is sparse so that the top-k operation for centers filtering is no longer necessary. Finally, each shifted *thing* voxel can be clustered to its closest center according to their spatial distance on the BEV space. We use the grid size 0.2m × 0.2m for our clustering pseudo heatmap in all the experiments.

Center grouping. By further analyzing the results of the offset regression, it is observed that clustering itself works worse for big objects such as buses than small objects such as cars and persons. The reason is that by LiDAR sensors, usually less body part can be scanned for a big object, especially when it is close to the LiDAR origin. As illustrated in Fig. 3 (a), the *thing* points of the bus are clustered into four instance IDs instead of one as expected. In other words, multiple centers originally belonging to the same instance may be generated from the pseudo heatmap.

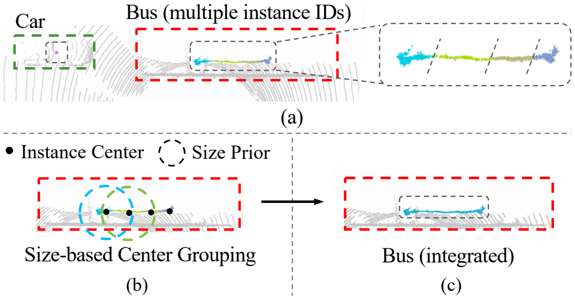


Figure 3. (a) illustrates a bad offset regression for a bus that is close to the origin of the LiDAR coordinate. The shifted *thing* points in different color represent different instance IDs. (b) and (c) show that with our center grouping module, the bus can be appropriately integrated.

To deal with such cases, we introduce a size-based center grouping module. We first use a 2D average pooling on the pseudo image $I' \in \mathbb{R}^{H\times W\times C_n}$ to count the number of *thing* voxels within a sliding window for each category. Majority voting is applied to determine the class of each grid. We then give each center a minimum radius empirically based on its category. Fig. 3 (b) illustrates the grouping operation: given a certain base center $C_b \in \mathcal{ID}_b$ as well as a radius r_b, where $\mathcal{ID}_b$ indicates a group of centers with the same instance ID as C_b. If a target center $C_t \in \mathcal{ID}_t$ with the same semantic label as C_b appears within the radius r_b, the set $\mathcal{ID}_t$ are then regrouped into $\mathcal{ID}_b$. We traverse all centers with this operation, through which multiple redundant centers are integrated together as shown in Fig. 3 (c). Since the number of instance centers per LiDAR frame is limited, the time cost can almost be ignored.

3.3. Knn-Transformer

At the beginning of the instance branch, we use the *thing* mask generated from semantic segmentation to pick out the feature vectors $F \in \mathbb{R}^{M\times C}$ for *thing* voxels. Since the number of the voxels to be processed is significantly reduced, accurately modeling the interaction among these elements becomes possible. Similar to a natural sentence, the disorderd and number-unfixed *thing* voxels are suitable for Transformer [31] to deal with.

Inspired by Swin-Transformer [21] where the local attention mechanism is introduced, we propose a knn-transformer to efficiently model the interaction among *thing* voxels. We basically follow the design of self-attention layer from [31], except that we take spatial distance as prior to construct the similarity matrix among local *thing* voxels. More specifically, given the features of *thing* voxels with shape $M \times C$ as shown in Fig. 4, we calculate the indices of k nearest neighbors for each *thing* voxel on GPU based on its spatial location, by which the input vectors are broadcasted to be a feature matrix with shape $M\times k\times C$. Through linear transformation, a query matrix Q, a key matirx K and a value matix V are generated respectively. Afterwards, an

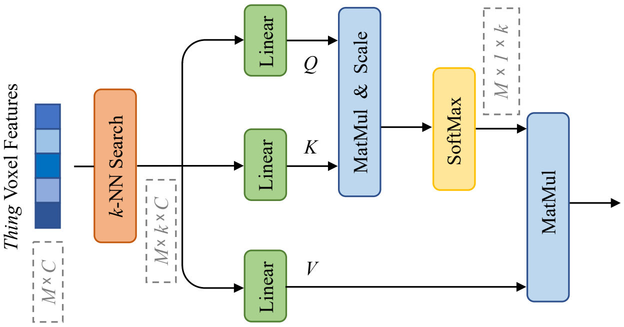


Figure 4. The self-attention layer in our knn-transformer module.

attention matrix with shape $M \times k$, describing the interaction between each *thing* voxel and its k nearest neighbors, is computed as [31]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{C'}}\right)V \quad (1)$$

where C' denotes the size of the channel dimension. Compared with the vanilla self-attention layer, our knn-based design reduces the computational complexity per layer from $O\left(M^2 \cdot C'\right)$ to $O\left(M \cdot k \cdot C'\right)$. We maintain the structure of multi-head attention and feed-forward layer in [31]. Since the position information is encoded in each voxel by nature, the positional embedding is not adopted in our module. We use $k = 25$ in our experiments.

3.4. Backbone Design

Space partition. In our 2D backbone network, we utilize the polar BEV coordinate for space partition based on two reasons. First, objects not only remain unchanged in scale but also rarely overlap [40] in the BEV space. Second, the distribution of points under different ranges can be balanced in the polar coordinate [37]. To facilitate the projection process from 3D to 2D space, cylindrical space partition [41] is adopted for voxel feature extraction.

Voxel encoder. Following [41], we first group a frame of raw LiDAR point cloud into a voxel representation with shape $\left(N \times N_p^*\right) \times K$ based on the location of each point in the cylindrical space, where K is the feature dimension of LiDAR points, N is the number of non-empty voxels and N_p^* denotes the different number of points in each voxel. A shared three-layer MLP with BatchNorm and ReLU is utilized to extract point features, followed by a max-pooling layer to create a consistent representation for each voxel. A single-layer MLP is adopted for feature reduction to generate the fine-grained voxel features with shape $N \times 16$.

BEV encoder and 2D backbone network. We futher encode features with different receptive fields in the 2D BEV space. On the one hand, the operations of interaction in 3D space are time-consuming and cost large memories. On the other hand, for a 2.5D scene scanned by LiDAR [15], it is unnecessary to extract features entirely in 3D space. Precisely, we first map the fine-grained voxel features $V \in \mathbb{R}^{N \times 16}$ to a polar BEV image $I_p' \in \mathbb{R}^{(H \times W \times N_v^*) \times 16}$, where H and W are the size of the BEV map and N_v^* is the different number of voxels in each BEV grid. A shared MLP is applied for feature extraction. Similar with the voxel encoder, we use a max-pooling layer at each BEV grid to create a consistent representation $I_p \in \mathbb{R}^{H \times W \times 64}$. Subsequently, a Unet-like 2D backbone network is adopted to encode features with different receptive fields in the BEV space as [40].

We have four decoders in our 2D backbone network, where the first two are shared for the semantic and instance branches. We gather the BEV features for the corresponding voxels in both branches respectively according to their BEV locations. The gathered BEV features are then concatenated with the fine-grained voxel features as the final voxel representation. All prediction results and supervision signals are at the voxel level. Finally, we map the voxel results into point level based on the coordinate of each point.

4. Experiments

We evaluate our proposed Panoptic-PHNet on both SemanticKITTI and nuScenes datasets. Due to page limitations, please refer to the supplementary material for more details on the experiments and qualitative results.

SemanticKITTI. SemanticKITTI [1] is the first dataset that presents challenges for LiDAR panoptic segmentation. It is derived from KITTI [12] odometry dataset, and contains 22 data sequences with a 64-beams LiDAR sensor, 10 of which are for training, 11 for testing and 1 for validation. There are annotated point-wise labels in 20 classes for segmentation tasks, 8 of which are defined as *thing* classes.

NuScenes. NuScenes [3] is a large-scale driving dataset with a wide diversity of urban scenes. It contains 1000 scenes of 20s duration. The annotations are created every 0.5s with a 32-beams LiDAR sensor. Recently, the official expanded the point-wise annotations for LiDAR panoptic segmentation task with 16 semantic classes, 10 of which are *thing* classes. Since no one has reported results on the nuScenes test server for this new task yet, we mainly compare our results with strong baselines reported by the official [10] on the test and validation sets.

Evaluation Metrics As defined in [18], we use panoptic quality (PQ), segmentation quality (SQ) and recognition quality (RQ) to evaluate panoptic segmentation. These metrics are calculated separately for *thing* and *stuff* classes indicated by PQ^{Th}, SQ^{Th}, RQ^{Th} and PQ^{St}, SQ^{St}, RQ^{St}. Following [25], we also report $\text{PQ}^{\dagger}$ to use SQ as PQ for *stuff* classes. We use mean IoU (mIoU) to evaluate the quality of semantic segmentation. In addition, we adopt average EPE (end-point-error) from the visual optical flow field as the

Method	PQ	PQ†	RQ	SQ	PQ$^{\text{Th}}$	RQ$^{\text{Th}}$	SQ$^{\text{Th}}$	PQ$^{\text{St}}$	RQ$^{\text{St}}$	SQ$^{\text{St}}$	mIoU	FPS
RangeNet++ [23] + PointPillars [20]	37.1	45.9	47.0	75.9	20.2	25.2	75.2	49.3	62.8	76.5	52.4	2.4
LPSAD [22]	38.0	47.0	48.2	76.5	25.6	31.8	76.8	47.1	60.1	76.2	50.9	11.8
KPConv [30] + PointPillars [20]	44.5	52.5	54.4	80.0	32.7	38.7	81.5	53.1	65.9	79.0	58.8	1.9
Panoster [11]	52.7	59.9	64.1	80.7	49.4	58.5	83.3	55.1	68.2	78.8	59.9	-
Panoptic-PolarNet [40]	54.1	60.7	65.0	81.4	53.3	60.6	87.2	54.8	68.1	77.2	59.5	11.6
DS-Net [14]	55.9	62.5	66.7	82.3	55.1	62.8	87.2	56.5	69.5	78.7	61.6	3.2†
EfficientLPS [28]	57.4	63.2	68.7	83.0	53.1	60.5	87.8	**60.5**	**74.6**	79.5	61.4	-
Panoptic-PHNet	**61.5**	**67.9**	**72.1**	**84.8**	**63.8**	**70.4**	**90.7**	59.9	73.3	**80.5**	**66.0**	11.0
Panoptic-PHNet §	64.6	70.2	74.9	85.7	66.9	73.3	91.5	63.0	76.1	81.5	68.4	-

Table 1. LiDAR panoptic segmentation resuts on the **test** set of SemanticKITTI. Metrics are in [%] and FPS is in [Hz]. (†: we measure the latency of [14] with official codebase released by the authors on our hardware for reference; §: our method with double-flip and multi-model ensemble.)

Method	PQ	PQ†	RQ	SQ	PQ$^{\text{Th}}$	RQ$^{\text{Th}}$	SQ$^{\text{Th}}$	PQ$^{\text{St}}$	RQ$^{\text{St}}$	SQ$^{\text{St}}$	mIoU
EfficientLPS [28]	62.4	66.0	74.1	83.7	57.2	68.2	83.6	71.1	84.0	83.8	66.7
Panoptic-PolarNet [40]	63.6	67.1	75.1	84.3	59.0	69.8	84.3	71.3	83.9	84.2	67.0
SPVNAS [29] + CenterPoint [36]	72.2	76.0	81.2	88.5	71.7	79.4	89.7	73.2	84.2	86.4	76.9
Cylinder3D++ [41] + CenterPoint [36]	76.5	79.4	85.0	89.6	76.8	84.0	91.1	76.0	**86.6**	87.2	77.3
(AF)^2-S3Net [6] + CenterPoint [36]	76.8	80.6	85.4	89.5	79.8	86.8	91.8	71.8	83.0	85.7	78.8
Panoptic-PHNet	**80.1**	**82.8**	**87.6**	**91.1**	**82.1**	**88.1**	**93.0**	**76.6**	**86.6**	**87.9**	**80.2**
Panoptic-PHNet §	81.5	84.0	88.4	91.9	83.5	88.7	93.9	78.2	87.8	88.6	81.5

Table 2. LiDAR panoptic segmentation resuts on the **test** set of nuScenes. All scores are in [%]. (§: our method with double-flip and multi-model ensemble.)

metric for offset regression to compare our approach with other clustering-based methods.

Training and Inference. We use the same configuration and training schedules as previous works [40, 41]. See supplementary material for detailed hyper-parameters. During training, we use the cross-entropy loss ($\mathcal{L}_{ce}$) and Lovasz softmax loss [2] ($\mathcal{L}_{ls}$) to train the semantic head. In the instance head, we use L1 loss ($\mathcal{L}_{l1}$) for the offset regression. The final loss is denoted as:

$$\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{ls} + \mathcal{L}_{l1} \tag{2}$$

As mentioned in [40], we also find that the number of dynamic instances in SemanticKITTI is limited, hence we adopt a copy-paste data augmentation scheme from [35] to alleviate the distribution imbalance among categories. On nuScenes, however, we do not use this data augmentation scheme, for the reason that each frame in nuScenes has 34 instances on average, which is 6 times more than that of SemanticKITTI and enough to drive the training of semantic head. During inference, we follow [40] to merge the results from two branches to generate the final panoptic segmentation predictions. The inference latency is measured on a platform with an Intel Core i7 CPU and a RTX 2080Ti GPU.

4.1. Main Results

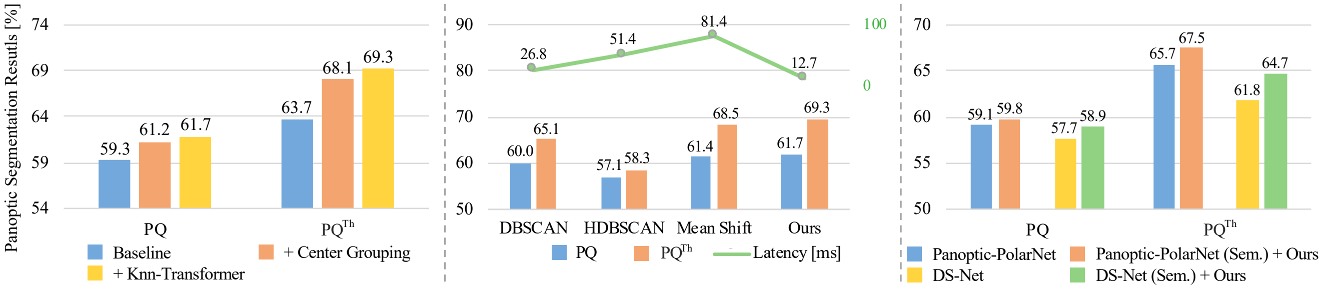
Results on SemanticKITTI. We first compare our method with the state-of-the-art LiDAR panoptic segmentation methods on SemanticKITTI test set. As shown in Tab. 1, our method outperforms all existing methods with remarkable margins, i.e., improving PQ by 4.1% (61.5% vs. 57.4%) and PQ$^{\text{Th}}$ by 8.7% (63.8% vs. 55.1%) with a real-time speed. Compared with 3D CNN based methods, such as DS-Net [14], our approach achieves higher accuracy and is more than 3 times faster (11 FPS vs. 3.2 FPS). In regard of the 2D CNN based methods, e.g., Panoptic-PolarNet [40], our method achieves over 10% promotion in PQ$^{\text{Th}}$ with a similar inference speed due to the combination of fine-grained voxel features and 2D CNN features. Moreover, following [36], we also report our test-time-augmentation (TTA) version including double-flip and multi-model ensemble for reference (last line of Tab. 1) to show the upper bound of our framework.

Results on NuScenes. Recently nuScenes released the test server for LiDAR panoptic segmentation along with the results of multiple strong baselines [10]. As shown in Tab. 2, our method surpasses the best baseline method by 3.3% PQ, 2.3% PQ$^{\text{Th}}$ and 4.8% PQ$^{\text{St}}$. Since the official combined approaches are obtained by downloading individual submissions from various evaluation servers, which may use incorporated TTA such as Cylinder3D++ [41], we also report our TTA version as we do on SemanticKITTI. Although the combined methods based on the powerful CenterPoint in Tab. 2 perform relatively better on nuScenes, we argue that such top-down approaches, depending heavily on detectors, are limited as analyzed in Sec. 1. Based on elaborative architecture design, our end-to-end framework still achieves leading results.

Following the recent official evaluation protocol [10], we further report our results on nuScenes validation as shown in Tab. 3 without any test-time augmentation techniques. Our approach outperforms the best baseline Panoptic-PolarNet [40] by 11.3% PQ and 14.8% PQ$^{\text{Th}}$.

Method	PQ	PQ†	RQ	SQ	PQ$^{\text{Th}}$	RQ$^{\text{Th}}$	SQ$^{\text{Th}}$	PQ$^{\text{St}}$	RQ$^{\text{St}}$	SQ$^{\text{St}}$	mIoU
PanopticTrackNet [17]	51.4	56.2	63.3	80.2	45.8	55.9	81.4	60.4	75.5	78.3	58.0
EfficientLPS [28]	62.0	65.6	73.9	83.4	56.8	68.0	83.2	70.6	83.6	83.8	65.6
Panoptic-PolarNet [40]	63.4	67.2	75.3	83.9	59.2	70.3	84.1	70.4	83.5	83.6	66.9
Panoptic-PHNet	**74.7**	**77.7**	**84.2**	**88.2**	**74.0**	**82.5**	**89.0**	**75.9**	**86.9**	**86.8**	**79.7**

Table 3. LiDAR panoptic segmentation resuts on nuScenes validation. All scores are in [%].



(a) Ablation Study of Network Components

(b) Ablation Study of Clustering Algorithms

(c) Ablation Study of Instance Segmentation

Figure 5. Ablation study on SemanticKITTI validation. (a) The network benefits from the two proposed components. (b) Our approach based on the clustering pseudo heatmap is faster and more accurate. (c) Fed with the same results of semantic segmentation respectively, our instance segmentation performs better than the two state-of-the-art LiDAR panoptic segmentation methods.

4.2. Ablation Study

Ablation on Network Components. We first analyze the proposed center grouping module for our clustering pseudo heatmap as well as the knn-transformer, which are enabled sequentially. The baseline result comes from our Panoptic-PHNet model without these two modules.

As shown in Fig. 5 (a), both of the two modules contribute to the final performance. Regarding the center grouping module, it presents a significant quality improvement for instance segmentation (+4.4% PQ$^{\text{Th}}$). As mentioned in Sec. 3.2, our clustering pseudo heatmap indeed achieves high recall for center generation: as long as there is a cluster of points, a highlight peak shows up certainly. However, it also brings the issue of multiple redundant centers. Solving this problem effectively, the proposed center grouping further ensures the high precision for instance centers. Note that a possible limitation of this module lies in the introduced hyper-parameters, i.e., the prior size for each category, which can be left in further work.

In addition, by enhancing the feature representations of *thing* voxels for more accurate offset regression, our knn-transformer further improves the performance by 1.2% PQ$^{\text{Th}}$, further analysis is shown later.

Ablation on Clustering Algorithms. We compare our clustering scheme, i.e., the center grouping based clustering pseudo heatmap, with the widely used heuristic clustering algorithms: DBSCAN [8], HDBSCAN [4] and Mean Shift [7] in a plug-in manner as [14]. Fig. 5 (b) shows our clustering scheme outperforms all the listed clustering algorithms in both accuracy and speed. In contrast with the Mean Shift that shows the best accuracy among the heuristic algorithms, ours is more than 6 times faster (12.7ms vs. 81.4ms). It is noteworthy that in terms of accuracy, our method is just 0.8% PQ$^{\text{Th}}$ higher than Mean Shift, which is quite different from the experiments of DS-Net [14]: their dynamic shifting clustering algorithm surpasses Mean Shift by 3.2% PQ$^{\text{Th}}$. The reason inside lies in the different bases of offset regression, our high-quality offset prediction itself leads to remarkable performance even with Mean Shift, which we explain in detail later.

Ablation on Instance Segmentation. We further figure out the performance of our instance segmentation compared with two clustering-based methods, i.e., DS-Net which is based on a dynamic shifting module to shift *thing* points towards the instance centers in an iterative manner, as well as Panoptic-PolarNet [40] requiring a heatmap by learning to yield instance centers. To eliminate the influence of the semantic segmentation, we replace the results of our semantic branch with theirs. In this way, our framework generates instance IDs for their *thing* points. As shown in Fig. 5 (c), our instance segmentation brings improvements of 1.8% and 2.9% in PQ$^{\text{Th}}$ for the two state-of-the-art methods.

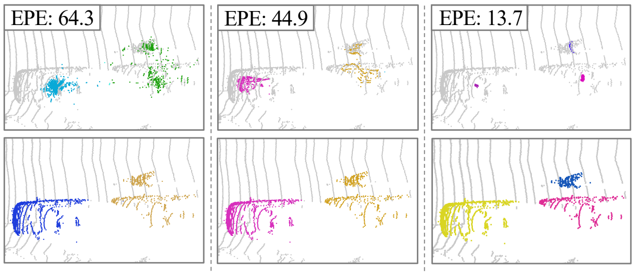
Ablation on Fine-grained Voxel Features. We separately verify the influence of the fine-grained voxel features on the semantic branch and the instance branch. In ablation for the semantic branch, we follow Panoptic-PolarNet [40] to use BEV features only to generate multiple predictions, which are reshaped back to voxels. As shown in Tab. 4, the use of fine-grained voxel features improves mIoU by 1.2%, and the improved mIoU further promotes PQ by 1.1% and PQ$^{\text{Th}}$ by 1.7%. It can be observed that the fine-grained voxel fea-

Branch	Method	PQ	PQ^{Th}	mIoU
Semantic	BEV Feature	60.6	67.6	64.5
	+ Voxel Feature	**61.7**	**69.3**	**65.7**
Instance	BEV Feature	61.6	69.0	-
	+ Voxel Feature	**61.7**	**69.3**	-

Table 4. Ablation on fine-grained voxel features. We show the experiments for the semantic branch and the instance branch respectively on SemanticKITTI validation.

Method	PQ	PQ^{Th}	mIoU	EPE ↓
DS-Net [14]	57.7	61.8	63.5	64.3
Panoptic-PolarNet [40]	59.1	65.7	64.5	44.9
Ours	**61.7**	**69.3**	**65.7**	**13.7**

Table 5. Comparison among clustering-based methods on SemanticKITTI validation. EPE is calculated by shifted *thing* points for each method. Panoptic metrics are in [%] and EPE is in [cm]. ↓ is for lower better.



DS-Net Panoptic-PolarNet Ours

Figure 6. Contrast of qualitative results. The top three images show the shifted *thing* points via the predicted offsets from different methods. The bottom images show the results of instance segmentation. Colored points represent different instance IDs. In this case, the three close instances are correctly segmented only with our method. Better viewed in color and zoomed in for details.

tures are crucial for the semantic segmentation task. As for the instance branch, the fine-grained voxel features are also beneficial although not so obvious as it is in the semantic branch. Overall, despite a little computational load, the combination of fine-grained voxel features and BEV features with different receptive fields brings our strong backbone network in terms of accuracy and speed for LiDAR panoptic segmentation.

4.3. Further Analysis

Effects of Offset Regression. We further analyze the accuracy of offset regression. It is actually a key factor in clustering-based panoptic segmentation methods, for which, however, related evaluation researches can rarely be found. Thus, we adopt the average EPE (end-point-error), which is used in the evaluation of the visual optical flow field, to validate the effects of offset regression.

We still compare our method with DS-Net [14] and Panoptic-PolarNet [40]. Since the latter only has 2D offset predictions, all the EPE results in Tab. 5 are calculated in the cartesian 2D BEV space for a fair comparison. As shown in the table, our approach has an obviously smaller offset error, which means larger spatial distance among instances to facilitate the following clustering process. Although the index of mIoU also contributes to PQ, it is clear that our network benefits from such high-precision offset predictions. Moreover, high-quality offset regression presents greater impact on crowded urban scenes as illustrated in Fig. 6, where only our approach segments three close instances correctly. Such crowded scenes are more common in nuScenes dataset, on which, thus, our method shows more significant improvement. As demonstrated in Tab. 3 and Tab. 5, for example, compared with Panoptic-PolarNet, the promotions of our approach are 14.8% PQ^{Th} vs. 3.6% PQ^{Th} on the two datasets respectively.

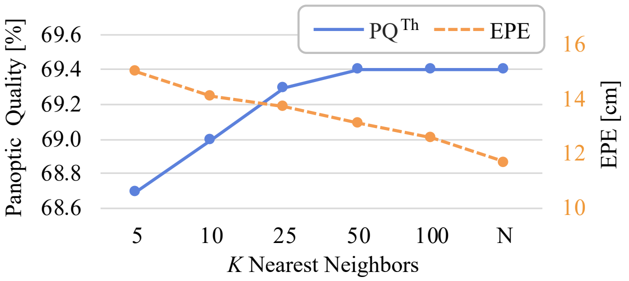


Figure 7. Effects of the selected k values in knn-transformer. The primary y-axis on the left is for PQ^{Th}, and the secondary y-axis on the right is for EPE.

Effects of Knn-Transformer. We evaluate the effects of a set of k values for our knn-transformer. As shown in Fig. 7, along with the increase of k, the PQ^{Th} improves at first, but almost saturates when k is greater than 25. The EPE of offset regression, however, decreases continuously, which proves the effectiveness of our knn-transformer module. Considering the memory footprint, we use $k = 25$ as the final choice.

5. Conclusion

In this paper, we propose a real-time and high-precision LiDAR panoptic segmentation framework named Panoptic-PHNet. As a new paradigm, we present a clustering pseudo heatmap, which is directly generated from shifted *thing* points without extra learning tasks to yield instance centers, followed by a center grouping module towards the issue of multiple redundant centers. A knn-transformer is introduced to efficiently model the interaction among *thing* voxels for feature enhancement to improve the accuracy of offset regression. Finally, based on a strong backbone design, which fuses the fine-grained voxel features and 2D BEV features with different receptive fields, our Panoptic-PHNet achieves the state-of-the-art performance on both SemanticKITTI and nuScenes datasets.

References

[1] Jens Behley, Andres Milioto, and Cyrill Stachniss. A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI. *arXiv preprint arXiv:2003.02371*, 2020. 1, 3, 5

[2] 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 *CVPR*, pages 4413–4421, 2018. 6

[3] 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 *CVPR*, pages 11618–11628, 2020. 5

[4] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*, pages 160–172, 2013. 1, 7

[5] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, pages 12472–12482, 2020. 2, 3

[6] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (AF)2-S3Net: Attentive Feature Fusion With Adaptive Feature Selection for Sparse Semantic Segmentation Network. In *CVPR*, pages 12547–12556, 2021. 6

[7] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 603–619, 2002. 1, 7

[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 7

[9] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. RangeDet: In Defense of Range View for LiDAR-based 3D Object Detection. *arXiv preprint arXiv:2103.10039*, 2021. 2

[10] 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. *arXiv preprint arXiv:2109.03805*, 2021. 1, 3, 5, 6

[11] 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 Autom. Lett.*, pages 3216–3223, 2021. 3, 6

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 5

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 3

[14] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *CVPR*, pages 13090–13099, 2021. 1, 3, 6, 7, 8

[15] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What You See is What You Get: Exploiting Visibility for 3D Object Detection. In *CVPR*, pages 10998–11006, 2020. 5

[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 *CVPR*, pages 11105–11114, 2020. 2

[17] Juana Valeria Hurtado, Rohit Mohan, and Abhinav Valada. MOPT: multi-object panoptic tracking. In *CVPR Workshop*, 2020. 1, 3, 7

[18] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic Segmentation. In *CVPR*, pages 9404–9413, 2019. 1, 5

[19] Jean Lahoud, Bernard Ghanem, Martin R. Oswald, and Marc Pollefeys. 3D Instance Segmentation via Multi-Task Metric Learning. In *ICCV*, pages 9255–9265, 2019. 1

[20] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 2, 6

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 4

[22] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. Lidar panoptic segmentation for autonomous driving. In *IROS*, pages 8505–8512, 2020. 1, 3, 6

[23] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *IROS*, pages 4213–4220, 2019. 2, 6

[24] Quang-Hieu Pham, Duc Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds With Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields. In *CVPR*, pages 8827–8836, 2019. 1

[25] Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *CVPR*, pages 8277–8286, 2019. 5

[26] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, pages 77–85, 2017. 2

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 2

[28] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation. *CoRR*, abs/2102.08009, 2021. 6, 7

[29] 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 *ECCV*, pages 685–702, 2020. 2, 6

[30] 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 *ICCV*, pages 6410–6419, 2019. 2, 6

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2, 4, 5

[32] Yue Wang, Alireza Fathi, Abhijit Kundu, David A. Ross, Caroline Pantofaru, Thomas A. Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *ECCV*, pages 18–34, 2020. 2

[33] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In *ICRA*, pages 1887–1893, 2018. 2

[34] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, pages 4376–4382, 2019. 2

[35] Yan Yan, Yuxing Mao, and Bo Li. SECOND: sparsely embedded convolutional detection. *Sensors*, page 3337, 2018. 2, 6

[36] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-Based 3D Object Detection and Tracking. In *CVPR*, pages 11784–11793, 2021. 6

[37] 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 *CVPR*, pages 9598–9607, 2020. 2, 5

[38] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds. In *CoRL*, pages 923–932, 2019. 2

[39] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *CVPR*, pages 4490–4499, 2018. 2

[40] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *CVPR*, pages 13194–13203, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[41] 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 *CVPR*, pages 9939–9948, 2021. 5, 6