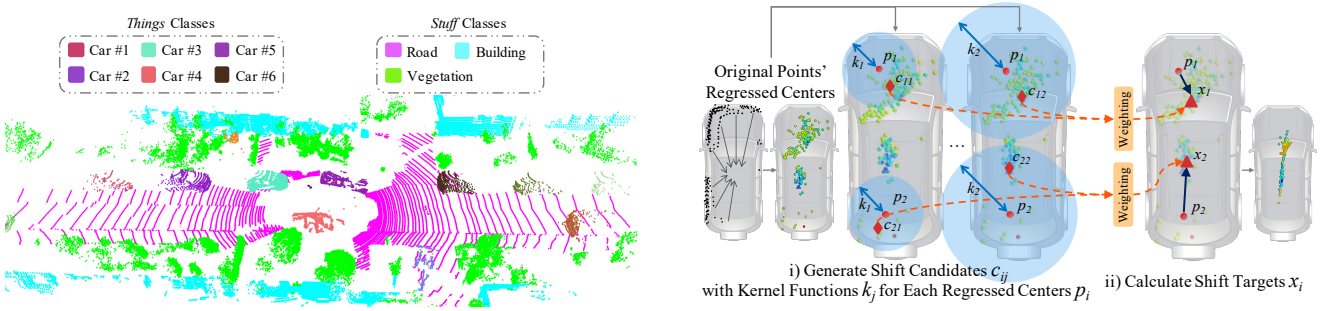
LiDAR-based Panoptic Segmentation via Dynamic Shifting Network

Fangzhou Hong[1,3] Hui Zhou[2] Xinge Zhu[3] Hongsheng Li[3,4] Ziwei Liu[1]✉
[1]S-Lab, Nanyang Technological University [2]Sensetime Research
[3]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[4]School of CST, Xidian University
{fangzhou001, ziwei.liu}@ntu.edu.sg zhouhui@sensetime.com
zx018@ie.cuhk.edu.hk hsli@ee.cuhk.edu.hk



(a) LiDAR-based Panoptic Segmentation (b) Dynamic Shifting

Figure 1: As shown in (a), LiDAR-based panoptic segmentation requires instance-level segmentation for *things* classes and semantic-level segmentation for *stuff* classes. (b) shows the core operation of the proposed dynamic shifting where several shift candidates are weighted to obtain the optimal shift target for each regressed center.

Abstract

*With the rapid advances of autonomous driving, it becomes critical to equip its sensing system with more holistic 3D perception. However, existing works focus on parsing either the objects (*e.g. *cars and pedestrians) or scenes (*e.g. *trees and buildings) from the LiDAR sensor. In this work, we address* ***the task of LiDAR-based panoptic segmentation****, which aims to parse both objects and scenes in a unified manner. As one of the first endeavors towards this new challenging task, we propose the Dynamic Shifting Network (DS-Net), which serves as an effective panoptic segmentation framework in the point cloud realm. In particular, DS-Net has three appealing properties:* ***1) strong backbone design.*** *DS-Net adopts the cylinder convolution that is specifically designed for LiDAR point clouds. The extracted features are shared by the semantic branch and the instance branch which operates in a bottom-up clustering style.* ***2) Dynamic Shifting for complex point distributions.*** *We observe that commonly-used clustering algorithms like BFS or DBSCAN are incapable of handling complex autonomous driving scenes with non-uniform point cloud distributions and varying instance sizes. Thus, we present an efficient learnable clustering module, dynamic shifting, which adapts kernel functions on-the-fly for different instances.* ***3) Consensus-driven Fusion.*** *Finally, consensus-driven fusion is used to deal with the disagreement between semantic and instance predictions. To comprehensively evaluate the performance of LiDAR-based panoptic segmentation, we construct and curate benchmarks from two large-scale autonomous driving LiDAR datasets, SemanticKITTI and nuScenes. Extensive experiments demonstrate that our proposed DS-Net achieves superior accuracies over current state-of-the-art methods. Notably, we achieve 1st place on the public leaderboard of SemanticKITTI, outperforming 2nd place by 2.6% in terms of the PQ metric* [1].

1. Introduction

Autonomous driving, one of the most promising applications of computer vision, has achieved rapid progress in recent years. Perception system, one of the most important modules in autonomous driving, has also attracted extensive studies in previous research works. Admittedly, the classic tasks of 3D object detection [18, 24, 32] and semantic segmentation [20, 30, 35] have developed relatively mature

[1]Accessed at 2020-11-16. Codes are available at https://github.com/hongfz16/DS-Net.

solutions that support real-world autonomous driving prototypes. However, there still exists a considerable gap between the existing works and the goal of holistic perception which is essential for the challenging autonomous driving scenes. In this work, we propose to close the gap by exploring the task of LiDAR-based panoptic segmentation, which requires full-spectrum point-level predictions.

Panoptic segmentation has been proposed in 2D detection [15] as a new vision task which unifies semantic and instance segmentation. Behley *et al*. [3] extend the task to LiDAR point clouds and propose the task of LiDAR-based panoptic segmentation. As shown in Fig. 1 (a), this task requires to predict point-level semantic labels for background (*stuff*) classes (*e.g*. road, building and vegetation), while instance segmentation needs to be performed for foreground (*things*) classes (*e.g*. car, person and cyclist).

Nevertheless, the complex point distributions of LiDAR data make it difficult to perform reliable panoptic segmentation. Most existing point cloud instance segmentation methods [10, 14] are mainly designed for dense and uniform indoor point clouds. Therefore, decent segmentation results can be achieved through the center regression and heuristic clustering algorithms. However, due to the non-uniform density of LiDAR point clouds and varying sizes of instances, the center regression fails to provide ideal point distributions for clustering. The regressed centers usually form noisy strip distributions that vary in density and sizes. As will be analyzed in Section 3.2, several heuristic clustering algorithms widely used in previous works cannot provide satisfactory clustering results for the regressed centers of LiDAR point clouds. To tackle the above mentioned technical challenges, we propose Dynamic Shifting Network (DS-Net) which is specifically designed for effective panoptic segmentation of LiDAR point clouds.

Firstly, we adopt a **strong backbone design** and provide a strong baseline for the new task. Inspired by [37], the cylinder convolution is used to efficiently extract grid-level features for each LiDAR frame in one pass which are further shared by the semantic and instance branches.

Secondly, we present a novel **Dynamic Shifting Module** designed to cluster on the regressed centers with complex distributions produced by the instance branch. As illustrated in Fig. 1 (b), the proposed dynamic shifting module shifts the regressed centers to the cluster centers. The shift targets x_i are adaptively computed by weighting across several shift candidates c_{ij} which are calculated through kernel functions k_j. The special design of the module makes the *shift* operation capable of dynamically adapting to the density or sizes of different instances and therefore shows superior performance on LiDAR point clouds. Further analysis also shows that the dynamic shifting module is robust and not sensitive to parameter settings.

Thirdly, the **Consensus-driven Fusion Module** is presented to unify the semantic and instance results to obtain panoptic segmentation results. The proposed consensus-driven fusion mainly solves the disagreement caused by the class-agnostic style of instance segmentation. The fusion module is highly efficient, thus brings negligible computation overhead.

Extensive experiments on SemanticKITTI demonstrate the effectiveness of our proposed DS-Net. To further illustrate the generalizability of DS-Net, we customize a LiDAR-based panoptic segmentation dataset based on nuScenes. As one of the first works for this new task, we present several strong baseline results by combining the state-of-the-art semantic segmentation and detection methods. DS-Net outperforms all the state-of-the-art methods on both benchmarks (1st place on the public leaderboard of SemanticKITTI).

The main contributions are summarized below: **1)** To our best knowledge, we present one of the first attempts to address the challenging task of LiDAR-based panoptic segmentation. **2)** The proposed DS-Net effectively handles the complex distributions of LiDAR point clouds, and achieves state-of-the-art performance on SemanticKITTI and nuScenes. **3)** Extensive experiments are performed on large-scale datasets. We adapt existing methods to this new task for in-depth comparisons. Further statistical analyses are carried out to provide valuable observations.

2. Related Works

Point Cloud Semantic Segmentation. According to the data representations of point clouds, most point cloud semantic segmentation methods can be categorized to point-based and voxel-based methods. Based on PointNet [22] and PointNet++ [23], KPConv [25], DGCNN [28], PointConv [31] and Randla-Net [13] can directly operate on unordered point clouds. However, due to space and time complexity, most point-based methods struggle on large-scale point clouds datasets *e.g*. ScanNet [9], S3DIS [1], and SemanticKITTI [2]. MinkowskiNet [7] utilizes the sparse convolutions to efficiently perform semantic segmentation on the voxelized large-scale point clouds. Different from indoor RGB-D reconstruct point clouds, LiDAR point clouds have non-uniform and sparse point distributions which require special designs of the network. SqueezeSeg [30] views LiDAR point clouds as range images while PolarNet [35] and Cylinder3D [37] divide the LiDAR point clouds under the polar and cylindrical coordinate systems.

Point Cloud Instance Segmentation. Previous works have shown great progress in the instance segmentation of indoor point clouds. A large number of point-based methods (*e.g*. SGPN [26], ASIS [27], JSIS3D [21] and JSNet [36]) split the whole scene into small blocks and learn point-wise embeddings for final clustering, which are limited by the heuristic post processing steps and the lack of perception.

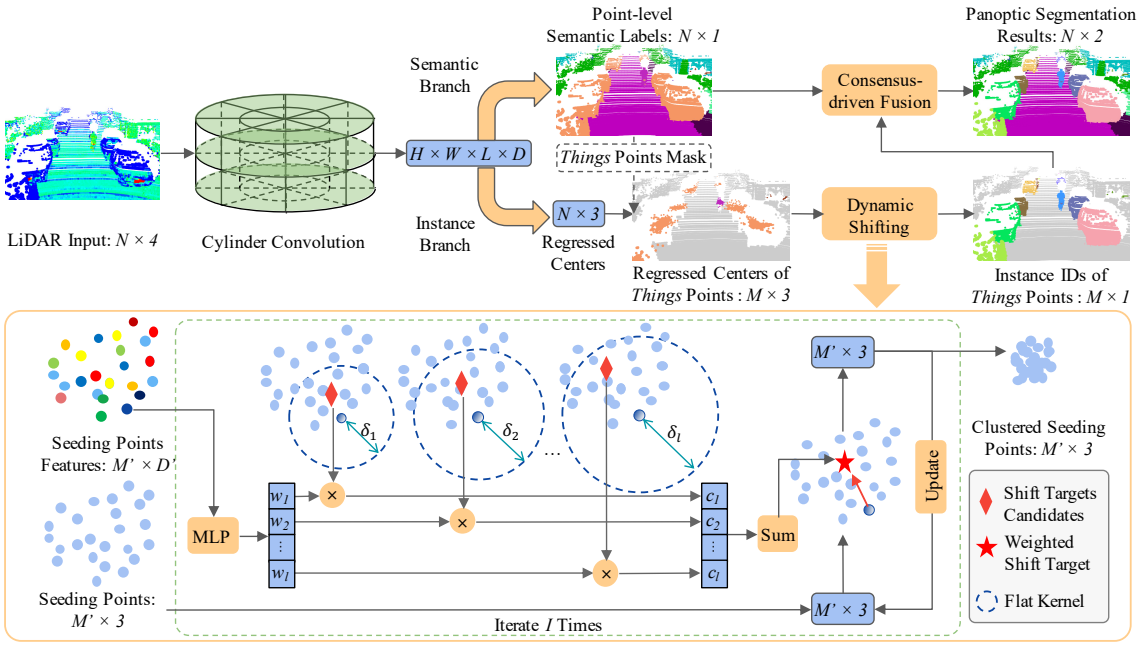


Figure 2: **Architecture of the DS-Net.** The DS-Net consists of the cylinder convolution, a semantic and an instance branch as shown in the upper part of the figure. The regressed centers provided by the instance branch are clustered by the novel dynamic shifting module which is shown in the bottom half. The consensus-driven fusion module unifies the semantic and instance results into the final panoptic segmentation results.

To avoid the problems, recent works (*e.g.* PointGroup [14], 3D-MPA [10], OccuSeg [12]) use sparse convolutions to extract features of the whole scene in one pass. As for LiDAR point clouds, there are a few previous works [13, 29, 30, 34] trying to tackle the problem. Wu *et al.* [30] directly cluster on XYZ coordinates after semantic segmentation. Wong *et al.* [29] builds the network in a top-down style and incorporates metric learning. Zhang *et al.* [34] uses grid-level center voting to cluster points of interest in autonomous driving scenes. Although our work is not targeting the instance segmentation for LiDAR point clouds, the proposed DS-Net can provide some insights into the challenging task.

3. Our Approach

As one of the first attempts on the task of LiDAR-based panoptic segmentation, we first introduce a strong backbone to establish a simple baseline (Sec. 3.1), based on which two modules are further proposed. The novel dynamic shifting module is presented to tackle the challenge of the non-uniform LiDAR point clouds distributions (Sec. 3.2). The efficient consensus-driven fusion module combines the semantic and instance predictions and produces panoptic segmentation results (Sec. 3.3). The whole pipeline of the DS-Net is illustrated in Fig. 2.

3.1. Strong Backbone Design

To obtain panoptic segmentation results, it is natural to solve two sub-tasks separately, which are semantic and instance segmentation, and combine the results. As shown in the upper part of Fig. 2, the strong backbone consists of three parts: the cylinder convolution, a semantic branch, and an instance branch. High quality grid-level features are extracted by the cylinder convolution from raw LiDAR point clouds and then shared by semantic and instance branches.

Cylinder Convolution. Considering the difficulty presented by the task, we find that the cylinder convolution [37] best meets the strict requirements of high efficiency, high performance and fully mining of 3D positional relationship. The cylindrical voxel partition can produce more even point distribution than normal Cartesian voxel partition and therefore leads to higher feature extraction efficiency and higher performance. Cylindrical voxel representation combined with sparse convolutions can naturally retain and fully explore 3D positional relationship. Thus we choose the cylinder convolution as our feature extractor.

Semantic Branch. The semantic branch performs semantic segmentation by connecting MLP to the cylinder convolution to predict semantic confidences for each voxel grid. Then the point-wise semantic labels are copied from their corresponding grids. We use the weighted cross entropy and Lovasz Loss [4] as the loss function of the semantic branch.

Instance Branch. The instance branch utilizes center regression to prepare the *things* points for further clustering. The center regression module uses MLP to adapt cylinder convolution features and make *things* points to regress the centers of their instances by predicting the offset vectors

$O \in \mathbb{R}^{M\times 3}$ pointing from the points $P \in \mathbb{R}^{M\times 3}$ to the instance centers $C_{gt} \in \mathbb{R}^{M\times 3}$. The loss function for instance branch can be formulated as:

$$L_{ins} = \frac{1}{M}\sum_{i=0}^{M}\|O[i] - (C_{gt}[i] - P[i])\|_1, \qquad (1)$$

where M is the number of *things* points. The regressed centers $O + P$ are further clustered to obtain the instance IDs, which can be achieved by either heuristic clustering algorithms or the proposed dynamic shifting module which are further introduced and analyzed in the following section.

3.2. Dynamic Shifting

Point Clustering Revisit. Unlike indoor point clouds which are carefully reconstructed using RGB-D videos, the LiDAR point clouds have the distributions that are not suitable for normal clustering solutions used by indoor instance segmentation methods. The varying instance sizes, the sparsity and incompleteness of LiDAR point clouds make it difficult for the center regression module to predict the precise center location and would result in noisy long strips distribution as displayed in Fig. 1 (b) instead of an ideal ball-shaped cluster around the center. Moreover, as presented in Fig. 3 (a), the clusters formed by regressed centers that are far from the LiDAR sensor have much lower densities than those of nearby clusters due to the non-consistent sparsity of LiDAR point clouds. Facing the non-uniform distribution of regressed centers, heuristic clustering algorithms struggle to produce satisfactory results. Four major heuristic clustering algorithms that are used in previous bottom-up indoor point clouds instance segmentation methods are analyzed below. The details of the following algorithms can be found in supplementary materials.

- **Breadth First Search (BFS).** BFS is simple and good enough for indoor point clouds as proved in [14], but not suitable for LiDAR point clouds. As discussed above, large density difference between clusters means that the *fixed radius* cannot properly adapt to different clusters. Small radius will over-segment distant instances while large radius will under-segment near instances.

- **DBSCAN [11] and HDBSCAN [6].** As density-based clustering algorithms, there is no surprise that these two algorithms also perform badly on the LiDAR point clouds, even though they are proved to be effective for clustering indoor point clouds [10, 33]. The core operation of DBSCAN is the same as that of BFS. While HDBSCAN intuitively assumes that the points with lower density are more likely to be noise points which is not the case in LiDAR points.

- **Mean Shift [8].** The advantage of Mean Shift, which is used by [17] to cluster indoor point clouds, is that the kernel function is not sensitive to density changes and robust to noise points which makes it more suitable than density-based algorithms. However, the *bandwidth* of the kernel function has great impact on the clustering results as shown in Fig. 3 (b). The fixed bandwidth cannot handle the situation of large and small instances simultaneously which makes Mean Shift also not the ideal choice for this task.

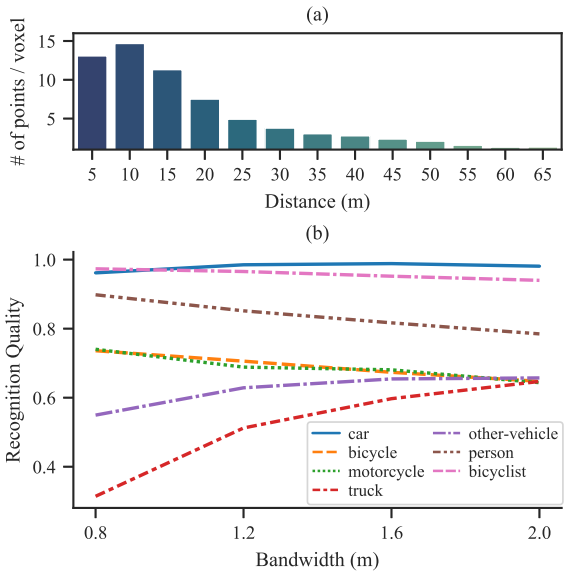


Figure 3: (a) counts the average number of regressed centers inside each valid voxel of instances at different distances. (b) shows the effect of Different Mean Shift Bandwidth on the Recognition Quality of Different Classes.

Dynamic Shifting. As discussed above, it is a robust way of estimating cluster centers of regressed centers by iteratively applying kernel functions as in Mean Shift. However, the fixed bandwidth of kernel functions fails to adapt to varying instance sizes. Therefore, we propose the dynamic shifting module which can automatically adapt the kernel function for each LiDAR point in the complex autonomous driving scene so that the regressed centers can be dynamically, efficiently and precisely shifted to the correct cluster centers.

In order to make the kernel function learnable, we first consider how to mathematically define a differentiable *shift* opration. Inspired by [16], the shift operation on the seeding points (*i.e.* points to be clustered) can be expressed as matrix operations if the number of iterations is fixed. Specifically, one iteration of shift operation can be formulated as follows. Denoting $X \in \mathbb{R}^{M\times 3}$ as the M seeding points, X will be updated once by the shift vector $S \in \mathbb{R}^{M\times 3}$ which is formulated as

$$X \leftarrow X + \eta S, \qquad (2)$$

where η is a scaling factor which is set to 1 in our experiments.

Algorithm 1: Forward Pass of the Dynamic Shifting Module

Input: *Things* Points $P \in \mathbb{R}^{M\times 3}$, *Things* Features $F \in \mathbb{R}^{M\times D'}$, *Things* Regressed Centers $C \in \mathbb{R}^{M\times 3}$, Fixed number of iteration $I \in \mathbb{N}$, Bandwidth candidates list $L \in \mathbb{R}^{l}$

Output: Instance IDs of *things* points $R \in \mathbb{R}^{M\times 1}$

1 $mask = FPS(P)$, $P' = P[mask]$
2 $X = C[mask]$, $F' = F[mask]$
3 **for** $i \leftarrow 1$ **to** I **do**
4 $W_i = Softmax(MLP(F'))$
5 $acc = zeros\_like(X)$
6 **for** $j \leftarrow 1$ **to** l **do**
7 $K_{ij} = (XX^T \leq L[j])$
8 $D_{ij} = diag(K_{ij}\mathbf{1})$
9 $acc = acc + W_i[:, j] \odot (D_{ij}^{-1}K_{ij}X)$
10 **end**
11 $X = acc$
12 **end**
13 $R' = cluster(X)$
14 $index = nearest\_neighbour(P, P')$
15 $R = R'[index]$
16 **return** R

The calculation of the shift vector S is by applying kernel function f on X, and formally defined as $S = f(X) - X$.

Among various kinds of kernel functions, the flat kernel is simple but effective for generating shift target estimations for LiDAR points, which is introduced as follows. The process of applying flat kernel can be thought of as placing a query ball of certain radius (*i.e.* bandwidth) centered at each seeding point and the result of the flat kernel is the mass of the points inside the query ball. Mathematically, the flat kernel $f(X) = D^{-1}KX$ is defined by the kernel matrix $K = (XX^T \leq \delta)$, which masks out the points within a certain bandwidth δ for each seeding point, and the diagonal matrix $D = diag(K\mathbf{1})$ that represents the number of points within the seeding point's bandwidth.

With a differentiable version of the shift operation defined, we proceed to our goal of dynamic shifting by adapting the kernel function for each point. In order to make the kernel function adaptable for instances with different sizes, the optimal bandwidth for each seeding point has to be inferenced dynamically. A natural solution is to directly regress bandwidth for each seeding point, which however is not differentiable if used with the flat kernel. Even though Gaussian kernel can make direct bandwidth regression trainable, it is still not the best solution as analyzed in section 4.1. Therefore, we apply the design of weighting across several bandwidth candidates to dynamically adapt to the optimal one.

One iteration of dynamic shifting is formally defined as follows. As shown in the bottom half of Fig. 2, l bandwidth candidates $L = \{\delta_1, \delta_2, ..., \delta_l\}$ are set. For each seeding point, l shift target candidates are calculated by l flat kernels with corresponding bandwidth candidates. Seeding points then dynamically decide the final shift targets, which are ideally the closest to the cluster centers, by learning the weights $W \in \mathbb{R}^{M\times l}$ to weight on l candidate targets. The weights W are learned by applying MLP and Softmax on the backbone features so that $\sum_{j=1}^{l} W[:, j] = \mathbf{1}$. The above procedure and the new learnable kernel function $\hat{f}$ can be formulated as

$$\hat{f}(X) = \sum_{j=1}^{l} W[:, j] \odot (D_j^{-1}K_jX), \tag{3}$$

where $K_j = (XX^T \leq \delta_j)$ and $D_j = diag(K_j\mathbf{1})$.

With the one iteration of dynamic shifting stated clearly, the full pipeline of the dynamic shifting module, which is formally defined in algorithm 1, can be illustrated as follows. Firstly, to maintain the efficiency of the algorithm, farthest point sampling (FPS) is performed on M *things* points to provide M' seeding points for the dynamic shifting iterations (Lines 1–2). After a fixed number I of dynamic shifting iterations (Lines 3–12), all seeding points have converged to the cluster centers. A simple heuristic clustering algorithm is performed to cluster the converged seeding points to obtain instance IDs for each seeding point (Line 13). Finally, all other *things* points find the nearest seeding points and the corresponding instance IDs are assigned to them (Lines 14–15).

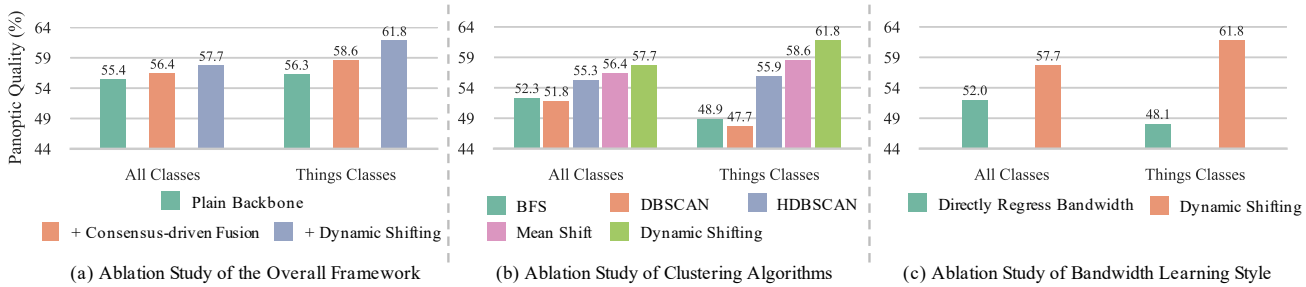
The optimization of dynamic shifting module is not intuitive since it is impractical to obtain the ground truth bandwidth for each seeding point. The loss function has to encourage seeding points shifting towards their cluster centers which have no ground truths but can be approximated by the ground truth centers of instances $C'_{gt} \in \mathbb{R}^{M'\times 3}$. Therefore, the loss function for the ith iteration of dynamic shifting is defined by the manhattan distance between the ground truth centers C'_{gt} and the ith dynamically calculated shift targets X_i, which can be formulated as

$$l_i = \frac{1}{M'} \sum_{x=1}^{M'} \|X_i[x] - C'_{gt}[x]\|_1. \tag{4}$$

Adding up all the losses of I iterations gives us the loss function L_{ds} for the dynamic shifting module:

$$L_{ds} = \sum_{i=1}^{I} w_i l_i, \tag{5}$$

where w_i are weights for losses of different iterations and are all set to 1 in our experiments.



(a) Ablation Study of the Overall Framework (b) Ablation Study of Clustering Algorithms (c) Ablation Study of Bandwidth Learning Style

Figure 4: **Ablation Study on the Validation Set of SemanticKITTI**. The proposed two modules both contributes to the final performance of the DS-Net. The dynamic shifting module has advantages in clustering LiDAR point clouds. Weighting on bandwidth candidates is better than directly regressing bandwidth.

3.3. Consensus-driven Fusion

Typically, solving the conflict between semantic and instance predictions is one of the essential steps in panoptic segmentation. The advantages of bottom-up methods are that all points with predicted instance IDs must be in *things* classes and one point will not be assigned to two instances. The only conflict needs to be solved is the disagreement of semantic predictions inside one instance, which is brought in by the class-agnostic way of instance segmentation. The strategy used in the proposed consensus-driven fusion is *majority voting*. For each predicted instance, the most appeared semantic label of its points determines the semantic labels for all the points inside the instance. This simple fusion strategy is not only efficient but could also revise and unify semantic predictions using instance information.

4. Experiments

We conduct experiments on two large-scale datasets: SemanticKITTI [2] and nuScenes [5].

SemanticKITTI. SemanticKITTI is the first dataset that presents the challenge of LiDAR-based panoptic segmentation and provides the benchmark[3]. SemanticKITTI contains 23,201 frames for training and 20,351 frames for testing. There are 28 annotated semantic classes which are remapped to 19 classes for the LiDAR-based panoptic segmentation task, among which 8 classes are *things* classes, and 11 classes are *stuff* classes. Each point is labeled with a semantic label and an instance id which will be set to 0 if the point belongs to *stuff* classes.

nuScenes. In order to demonstrate the generalizability of DS-Net, we construct another LiDAR-based panoptic segmentation dataset from nuScenes. With the point-level semantic labels from the newly released nuScenes *lidarseg* challenge and the bounding boxes provided by the detection task, we could generate instance labels by assigning instance IDs to points inside bounding boxes. Following the definition of the nuScenes *lidarseg* challenge, we mark 10 foreground classes as *things* classes and 6 background classes as *stuff* classes out of all 16 semantic classes. The training and validation set has 28,130 and 6,019 frames.

Evaluation Metrics. As defined in [3], the evaluation metrics of LiDAR-based panoptic segmentation are the same as that of image panoptic segmentation defined in [15] including Panoptic Quality (PQ), Segmentation Quality (SQ) and Recognition Quality (RQ) which are calculated across all classes. The above three metrics are also calculated separately on *things* and *stuff* classes which give PQ^{Th}, SQ^{Th}, RQ^{Th}, and PQ^{St}, SQ^{St}, RQ^{St}. $PQ^{\dagger}$ is defined by swapping PQ of each *stuff* class to its IoU then averaging over all classes. In addition, mean IoU (mIoU) is also used to evaluate the quality of the sub-task of semantic segmentation.

4.1. Ablation Study

Ablation on Overall Framework. To study on the effectiveness of the proposed modules, we sequentially add consensus-driven fusion module and dynamic shifting module to the bare backbone. The corresponding PQ and PQ^{Th} are reported in Fig. 4 (a) which shows that both modules contribute to the performance of DS-Net. The novel dynamic shifting module mainly boosts the performance of instance segmentation which are indicated by PQ^{Th} where the DS-Net outperforms the backbone (with fusion module) by 3.2% in validation split.

Ablation on Clustering Algorithms. In order to validate our previous analyses of clustering algorithms, we swap the dynamic shifting module for four other widely-used heuristic clustering algorithms: BFS, DBSCAN, HDBSCAN, and Mean Shift. The results are shown in Fig. 4 (b). Consistent with our analyses in Sec. 3.2, the density-based clustering algorithms (*e.g.* BFS, DBSCAN, HDBSCAN) perform badly in terms of PQ and PQ^{Th} while Mean Shift leads to the best results among the heuristic algorithms. Moreover, our dynamic shifting module shows the superiority over all four heuristic clustering algorithms.

Ablation on Bandwidth Learning Styles. In the dynamic shifting module, it is natural to directly regress bandwidth for each point as mentioned in Sec. 3.2. However, as shown in the Fig. 4 (c), direct regression is hard to optimize in this case because the learning target is not straightforward. It is difficult to determine the best bandwidth for each point, and therefore impractical to directly apply supervision on the

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

Method	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}	mIoU
KPConv [25] + PV-RCNN [24]	51.7	57.4	63.1	**78.9**	46.8	56.8	**81.5**	**55.2**	**67.8**	**77.1**	63.1
Cylinder3D [37] + PV-RCNN [24]	51.9	57.5	63.8	74.2	48.5	59.5	70.2	54.3	66.9	**77.1**	62.9
PointGroup [14]	46.1	54.0	56.6	74.6	47.7	55.9	73.8	45.0	57.1	75.1	55.7
LPASD [19]	36.5	46.1	-	-	-	28.2	-	-	-	-	50.7
DS-Net	**57.7**	**63.4**	**68.0**	77.6	**61.8**	**68.8**	78.2	54.8	67.3	**77.1**	**63.5**

Table 2: LiDAR-based panoptic segmentation results on the test set of SemanticKITTI. All results in [%]. "$*$" denotes the unpublished method which is in the 2nd place on the public benchmark of SemanticKITTI (accessed on 2020-11-16).

Method	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}	mIoU
KPConv [25] + PointPillars [18]	44.5	52.5	54.4	80.0	32.7	38.7	81.5	53.1	65.9	79.0	58.8
RangeNet++ [20] + PointPillars [18]	37.1	45.9	47.0	75.9	20.2	25.2	75.2	49.3	62.8	76.5	52.4
KPConv [25] + PV-RCNN [24]	50.2	57.5	61.4	80.0	43.2	51.4	80.2	55.9	68.7	**79.9**	**62.8**
LPASD [19]	38.0	47.0	48.2	76.5	25.6	31.8	76.8	47.1	60.1	76.2	50.9
PolarNet_seg*	53.3	59.8	64.2	81.1	52.1	59.5	86.9	54.2	67.6	76.9	58.9
DS-Net	**55.9**	**62.5**	**66.7**	**82.3**	**55.1**	**62.8**	**87.2**	**56.5**	**69.5**	78.7	61.6

regressed bandwidth. Therefore, it is easier for the network to choose from and combine several bandwidth candidates.

4.2. Evaluation Comparisons on SemanticKITTI

Comparison Methods. Since its one of the first attempts on LiDAR-based panoptic segmentation, we provide several strong baseline results in order to validate the effectiveness of DS-Net. As proposed in [3], one good way of constructing strong baselines is to take the results from semantic segmentation methods and detection methods, and generate panoptic segmentation results by assigning instance IDs to all points inside predicted bounding boxes. [3] has provided the combinations of KPConv [25] + PointPillars [18], and RangeNet++ [20] + PointPillars [18]. To make the baseline stronger, we combine KPConv [25] with PV-RCNN [24] which is the state-of-the-art 3D detection method. In addition to the above baselines, we also adapt the state-of-the-art indoor instance segmentation method PointGroup [14] using the official released codes to experiment on SemanticKITTI. Moreover, LPASD [19], which is one of the earliest works in this area, is also included for comparison.

Evaluation Results. Table 1 and 2 shows that the DS-Net outperforms all baseline methods in both validation and test splits by a large margin. The DS-Net surpasses the best baseline method KPConv + PV-RCNN in most metrics and especially has the advantage of 6% and 15% in terms of PQ and PQ^{Th} in validation split. In test split, the DS-Net outperforms KPConv + PV-RCNN by 5.7% and 11.9% in PQ and PQ^{Th}. On the leaderboard provided by [3], our DS-Net achieves 1st place and surpasses 2nd method "PolarNet_seg" by 2.6% and 3.0% in PQ and PQ^{Th} respectively. It is worth noting that PointGroup [14] performs poorly on the LiDAR point clouds which shows that indoor solutions are not suitable for challenging LiDAR point clouds. Further detailed results on SemanticKITTI can be found in supplementary materials.

4.3. Evaluation Comparisons on nuScenes

Comparison Methods. Similarly, two strong semantic segmentation + detection baselines are provided for comparison on nuScenes. The semantic segmentation method is Cylinder3D [37] and the detection methods are SECOND [32] and PointPillars [18]. For fair comparison, the detection networks are trained using single frames on nuScenes. The point-wise semantic predictions and predicted bounding boxes are merged in the following steps. First all points inside each bounding box are assigned a unique instance IDs across the frame. Then to unify the semantic predictions inside each instance, we assign the class labels of bounding boxes predicted by the detection network to corresponding instances.

Evaluation Results. As shown in Table 3, our DS-Net outperforms the best baseline method in most metrics. Especially, we surpass the best baseline method by 2.4% in PQ and 3.5% in PQ^{Th}. Unlike SemanticKITTI, nuScenes is featured as extremely sparse point clouds in single frames which adds even more difficulties to panoptic segmentation. The results validate the generalizability and the effectiveness of our DS-Net.

4.4. Further Analysis

Robust to Parameter Settings. As shown in Table 4, six sets of bandwidth candidates are set for independent training and the corresponding results are reported. The stable results show that DS-Net is robust to different parameter settings as long as the picked bandwidth candidates are comparable to the instance sizes. Unlike previous heuristic clustering algorithms that require massive parameter adjustment, DS-Net can automatically adjust to different instance sizes and point distributions and remains stable clustering quality. Further analyses on the iteration number settings are shown in supplementary materials.

Interpretable Learned Bandwidths. By averaging the

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

Method	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}	mIoU
Cylinder3D [37] + PointPillars [18]	36.0	44.5	43.0	83.3	23.3	27.0	83.7	57.2	69.6	82.7	52.3
Cylinder3D [37] + SECOND [32]	40.1	48.4	47.3	**84.2**	29.0	33.6	**84.4**	58.5	70.1	83.7	58.5
DS-Net	**42.5**	**51.0**	**50.3**	83.6	**32.5**	**38.3**	83.1	**59.2**	**70.3**	**84.4**	**70.7**

Table 4: Results of different bandwidth candidates settings. All results in [%].

Bandwidth Candidates (m)	PQ	PQ†	RQ	SQ	mIoU
0.2, 1.1, 2.0	57.4	63.0	67.7	77.4	**63.7**
0.2, 1.3, 2.4	57.5	63.1	67.7	**77.6**	63.5
0.2, 1.5, 2.8	57.6	63.2	67.8	**77.6**	**63.7**
0.2, 1.7, 3.2	**57.7**	**63.4**	**68.0**	**77.6**	63.5
0.2, 1.9, 3.6	**57.7**	63.3	67.9	**77.6**	63.4
0.2, 2.1, 4.0	57.4	63.1	67.7	77.5	63.3

bandwidth candidates weighted by the learned weights, the learned bandwidths for every points could be approximated. The average learned bandwidths of different classes are shown in Fig. 5. The average learned bandwidths are roughly proportional to the instance sizes of corresponding classes, which is consistent with the expectation that dynamic shifting can dynamically adjust to different instance sizes.

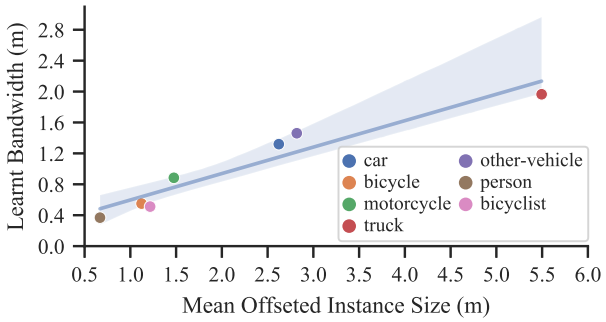


Figure 5: **Proportional Relationship Between Sizes and the Learned Bandwidths.** The x-axis represents the class-wise average size of regressed centers of instances while the y-axis stands for the average learned bandwidth of different *things* classes.

Visualization of Dynamic Shifting Iterations. As visualized in Fig. 6, the black points are the original point clouds of different instances including person, bicyclist and car. The seeding points are colored in spectral colors where the redder points represents higher learned bandwidth and bluer points represents lower learned bandwidth. The seeding points farther away from the instance centers tend to learn higher bandwidths in order to quickly converge. While the well-learned regressed points tend to have lower bandwidths to maintain their positions. After four iterations, the seeding points have converged around the instance centers.

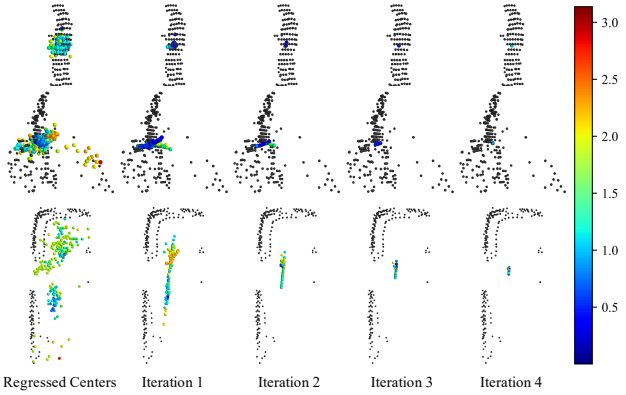


Figure 6: **Visualization of Dynamic Shifting Iterations.** The black points are the original LiDAR point clouds of instances. The colored points are seeding points. From left to right, with the iteration number increases, the seeding points converge to cluster centers.

5. Conclusion

With the goal of providing holistic perception for autonomous driving, we are one of the first to address the task of LiDAR-based panoptic segmentation. In order to tackle the challenge brought by the non-uniform distributions of LiDAR point clouds, we propose the novel DS-Net which is specifically designed for effective panoptic segmentation of LiDAR point clouds. Our DS-Net adopts strong baseline design which provides strong support for the consensus-driven fusion module and the novel dynamic shifting module. The novel dynamic shifting module adaptively shifts regressed centers of instances with different density and varying sizes. The consensus-driven fusion efficiently unifies semantic and instance results into panoptic segmentation results. The DS-Net outperforms all strong baselines on both SemanticKITTI and nuScenes. Further analyses show the robustness of the dynamic shifting module and the interpretability of the learned bandwidths.

Acknowledgments This research was conducted in collaboration with SenseTime. This work is supported by NTU NAP and A*STAR through the Industry Alignment Fund - Industry Collaboration Projects Grant. This work is supported in part by Centre for Perceptual and Interactive Intelligence Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants (Nos. 14208417 and 14207319), in part by CUHK Strategic Fund.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016. 2

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9297–9307, 2019. 2, 6

[3] Jens Behley, Andres Milioto, and Cyrill Stachniss. A benchmark for lidar-based panoptic segmentation based on kitti. *arXiv preprint arXiv:2003.02371*, 2020. 2, 6, 7

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

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 6

[6] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. 4

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

[8] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 4

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 2

[10] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9031–9040, 2020. 2, 3, 4

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 4

[12] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2940–2949, 2020. 3

[13] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 2, 3

[14] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020. 2, 3, 4, 7

[15] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9404–9413, 2019. 2, 6

[16] Shu Kong and Charless C Fowlkes. Recurrent pixel embedding for instance grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9018–9028, 2018. 4

[17] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9256–9266, 2019. 4

[18] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1, 7, 8

[19] A. Milioto, J. Behley, C. McCool, and C. Stachniss. Lidar panoptic segmentation for autonomous driving. In *IROS*, 2020. 7

[20] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. 1, 7

[21] Quang-Hieu Pham, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2019. 2

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

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

[24] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1, 7

[25] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J

Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 2, 7

[26] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. 2

[27] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019. 2

[28] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[29] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *The Conference on Robot Learning (CORL)*, 2019. 3

[30] 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 *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018. 1, 2, 3

[31] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 2

[32] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 7, 8

[33] Dongsu Zhang, Junha Chun, Sang Kyun Cha, and Young Min Kim. Spatial semantic embedding network: Fast 3d instance segmentation with deep metric learning. *arXiv preprint arXiv:2007.03169*, 2020. 4

[34] Feihu Zhang, Chenye Guan, Jin Fang, Song Bai, Ruigang Yang, Philip Torr, and Victor Prisacariu. Instance segmentation of lidar point clouds. *ICRA, Cited by*, 4(1), 2020. 3

[35] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 1, 2

[36] Lin Zhao and Wenbing Tao. Jsnet: Joint instance and semantic segmentation of 3d point clouds. In *AAAI*, pages 12951–12958, 2020. 2

[37] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and D. Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. *ArXiv*, abs/2011.10033, 2020. 2, 3, 7, 8