Unified 3D and 4D Panoptic Segmentation via Dynamic Shifting Networks

Fangzhou Hong, Lingdong Kong, Hui Zhou, Xinge Zhu, Hongsheng Li, Ziwei Liu✉

Abstract—With the rapid advances in autonomous driving, it becomes critical to equip its sensing system with more holistic 3D perception. However, widely explored tasks like 3D detection or point cloud semantic segmentation focus on parsing either the objects (*e.g.* cars and pedestrians) or scenes (*e.g.* trees and buildings). In this work, we propose to address the challenging task of **LiDAR-based Panoptic Segmentation**, which aims to parse both objects and scenes in a unified manner. In particular, we propose **Dynamic Shifting Network (DS-Net)**, which serves as an effective panoptic segmentation framework in the point cloud realm. DS-Net features a dynamic shifting module for complex LiDAR point cloud 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. To further explore the temporal information, we extend the single-scan processing framework to its temporal version, namely **4D-DS-Net**, for the task of **4D Panoptic Segmentation**, where the same instance across multiple frames should be given the same ID prediction. Instead of naïvely appending a tracking module to DS-Net, we propose to solve the 4D panoptic segmentation in a more unified way. Specifically, 4D-DS-Net first constructs 4D data volume by aligning consecutive LiDAR scans, upon which the temporally unified instance clustering is performed to obtain the final results. Extensive experiments on two large-scale autonomous driving LiDAR datasets, SemanticKITTI and Panoptic nuScenes, are conducted to demonstrate the effectiveness and superior performance of the proposed solution. The code is publicly available at https://github.com/hongfz16/DS-Net.

Index Terms—LiDAR Panoptic Segmentation, Point Cloud Semantic & Instance Segmentation, 4D Panoptic Segmentation.

1 Introduction

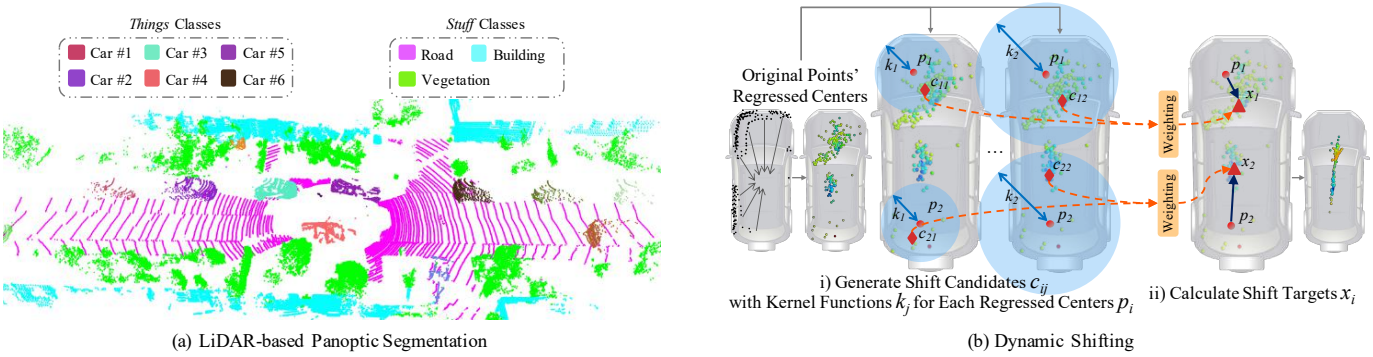
Autonomous driving, one of the most promising applications of computer vision, has achieved rapid progress in recent years. The perception system, one of the most important modules in autonomous driving, has also attracted extensive studies in previous research works. Admittedly, the classic tasks of 3D object detection [1], [2], [3] and semantic segmentation [4], [5], [6], [7], [8] have developed mature solutions that support real-world autonomous driving prototypes. However, there still exists a considerable gap between these tasks 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 3D and 4D panoptic segmentation, which requires dense point-level predictions in the spatial-temporal domain.

Panoptic segmentation for images [9] and videos [10] have been proposed as new vision tasks that unify semantic and instance segmentation. Behley *et al*. [11] extend the task to LiDAR point clouds and propose the task of LiDAR-based panoptic segmentation. Its temporal counterpart is also introduced as 4D panoptic segmentation [12] for more coherent perception in the temporal perspective. As shown in Fig. 1 (a), LiDAR-based panoptic segmentation 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 the foreground (*things*) classes (*e.g.* car, person and cyclist). The 4D panoptic segmentation further requires the same instance across different frames should be assigned with the same ID. Both tasks pose challenges from spatial and temporal perspectives, which are discussed as follows.

From the **spatial perspective**, complex point distributions of LiDAR scans make it difficult to perform reliable panoptic segmentation. Most existing point cloud instance segmentation methods [13], [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 size. As will be analyzed in Sec. 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 [15], the cylinder convolution is used to efficiently extract grid-level features for each LiDAR frame in one pass which are further

- *Fangzhou Hong and Ziwei Liu are with S-Lab, Nanyang Technological University, Singapore. E-mail: {fangzhou001,ziwei.liu}@ntu.edu.sg.*
- *Lingdong Kong is with the School of Computing, National University of Singapore. Email: lingdong@comp.nus.edu.sg.*
- *Hui Zhou is with SenseTime Research. E-mail: smarthuizhou@gmail.com.*
- *Xinge Zhu and Hongsheng Li are with the Chinese University of Hong Kong. E-mail: zx018@ie.cuhk.edu.hk, hsli@ee.cuhk.edu.hk.*
- ✉ *Corresponding Author*



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

Fig. 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.

shared by the semantic and instance branches.

Secondly, we present to regress the instance center for each point in the instance branch. Then we design a novel Dynamic Shifting Module to cluster on the regressed centers, which normally have complex distributions. As illustrated in Fig. 1 (b), the proposed dynamic shifting module shifts the regressed centers p_i to the cluster centers x_i. The cluster centers x_i are adaptively computed by weighting across several neighboring points 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.

From the **temporal perspective**, it is not trivial to effectively associate instances across frames, especially when they are moving and partially observed [16]. Sparse LiDAR point clouds give very few clues about their appearance, which makes it hard to distinguish them from each other. Therefore, it is sub-optimal to perform tracking based on multiple single-scan segmentation results, as proved in Sec. 4.4. Instead, we propose to deal with the 4D panoptic segmentation in a more unified way such that information from consecutive frames is fully utilized and associated implicitly.

Similar to 4D-PLS [12], we take poses estimated from the SLAM system to align and overlap consecutive LiDAR scans to form 4D data volumes. Then the temporally unified instance clustering is designed to perform instance segmentation in a frame-agnostic way, the results of which are further fused with the semantic segmentation to form the final 4D panoptic segmentation results. Such unified segmentation on the 4D data volume avoids the need for complicated post-tracking modules. In the meantime, the information extraction process is fully spatial-temporal-aware, making it more effective than "segment-then-track" approaches.

Extensive experiments on two large-scale autonomous driving datasets, SemanticKITTI [17] and Panoptic nuScenes [18], [19], demonstrate the effectiveness of our proposed DS-Net and 4D-DS-Net on LiDAR-based 3D and 4D panoptic segmentation. To further show the challenges of these tasks, we also present several strong baseline results by combining state-of-the-art semantic segmentation, detection, and tracking methods. Both DS-Net and 4D-DS-Net present competitive results on all testing benchmarks. More analyses are also conducted to provide more in-depth insights into the dynamic shifting module.

The main contributions are summarized below:

- The proposed DS-Net effectively handles the complex distributions of LiDAR point clouds and achieves competitive performance on the large-scale SemanticKITTI and Panoptic nuScenes datasets.
- We show a simple but effective solution to 4D panoptic segmentation by proposing 4D-DS-Net. Together with DS-Net, we formulate a unified 3D and 4D panoptic segmentation framework.
- Extensive experiments are performed to demonstrate effectiveness. Further statistical analyses are carried out to provide valuable observations.

2 Related Work

Image Panoptic Segmentation. The challenging vision task of panoptic segmentation was firstly defined by [9] where semantic segmentation for *stuff* classes [20] and instance segmentation for *things* classes are evaluated under unified metrics. From the perspective of network architecture, most panoptic segmentation methods can be categorized into top-down style and bottom-up style. The top-down methods are mostly based on MaskRCNN [21] where the instances are firstly detected and then segmented by predicting masks. The main innovations of this kind of method lie in the following two aspects. The first one [22], [23], [24], [25] is the backbone where semantic and instance information is extracted and shared. Panoptic FPN [22] and Seamless Scene Segmentation [23] manage to share MaskRCNN Feature Pyramid Network (FPN) between semantic and instance branches which yield a solid and strong baseline for the emerging task. The second aspect [22], [26], [27], [28] is the handling of disagreement between semantic and instance segmentation predictions and conflicts between multiple instance segmentation predictions. UPSNet [29] and Li *et al*. [30] try to unify *things* and *stuff* segmentation by introducing panoptic logits which can generate coherent panoptic segmentation results without any post-processing.

The bottom-up approaches [31], [32], [33] typically perform semantic segmentation first, then perform pixel clustering based on the semantic predictions, which naturally saves the trouble of conflict handling and would lead to lighter network design. Although top-down approaches tend to outperform bottom-up approaches due to the use of the powerful MaskRCNN, recent work of Panoptic-DeepLab [32] presents a bottom-up baseline that has comparable performance with top-down methods. For the simplicity of the network design, we choose to use a bottom-up approach in the proposed DS-Net.

Video Panoptic Segmentation. With the development of single-frame panoptic segmentation, recent research has extended the task to video inputs. Video panoptic segmentation requires the consistency of things IDs across frames [10], [34], [35], [36]. [10] constructs the method based on UPSNet [29]. Consequently, two frames are fused using spatial-temporal attention. Finally, object-level tracking is performed for consistent things IDs. Differently, [37] adopts a bottom-up backbone. Center regression is performed for two consequent frames with the centers predicted from the first frame, which would naturally produce consistent things IDs.

Point Cloud Semantic Segmentation. According to the data representations of point clouds, most point cloud semantic segmentation methods can be categorized into point-based and voxel-based methods. Based on PointNet-like backbones [38], [39], [40], KPConv [41], DGCNN [42], PointConv [43], Randla-Net [44], Pointasnl [45] 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 [46], S3DIS [47], and SemanticKITTI [17]. MinkowskiNet [48] and $(AF)^2$-S3Net [49] utilize the sparse convolutions to efficiently perform semantic segmentation on the voxelized large-scale point clouds. SqueezeSeg [5], [50], [51], [52], [53] views LiDAR point clouds as range images while PolarNet [6] and Cylinder3D [15], [54], [55] divide the LiDAR point clouds under the polar and cylindrical coordinate systems. Some works [56], [57], [58], [59] combine the advantages of point-based and voxel-based models to improve the semantic segmentation scores. Most recently, there are works that start to probe the data efficiency [60], generalizability [61], [62], and robustness [63], [64], [65], [66], [67] of this point cloud scene understanding task.

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 [68], ASIS [69], JSIS3D [70] and JSNet [71]) 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. To avoid the problems, recent works (*e.g.* PointGroup [14], 3D-MPA [13], OccuSeg [72]) use sparse convolutions to extract features of the whole scene in one pass. As for LiDAR point clouds, there are a few previous works [5], [44], [73], [74] trying to tackle the problem.

Point Cloud Panoptic Segmentation. Recently, many attempts have been made in point cloud panoptic segmentation. We categorize all LiDAR-based panoptic segmentation methods in Tab. 1. [11] first formally defines the task of LiDAR-based panoptic segmentation and proposes to combine semantic segmentation and 3D object detection to obtain the panoptic segmentation results. Most works can be categorized by the 3D representations (*e.g.* Point Cloud [41], Range View [5], BEV [1], [6], Cylinder3D [54], Voxels [48]) and segmentation styles (*e.g.* bottom-up and top-down). Utilizing strong point-based encoder-decoder structures (*e.g.* KPConv [41]), [12], [75], [76] performs panoptic segmentation directly on point clouds. [77], [78], [79], [80] perform spherical projection on LiDAR point clouds to form range views [5] and utilize 2D panoptic segmentation methods. [81], [82], [83] utilize the Polar BEV encoder [6] to extract per-point features. To further explore the contexts in 3D, many works [84], [85], [86], [87] use Cylinder3D [54] as the basic building block to extract per-point features. Some [88], [89], [90], [91], [92] also perform sparse convolution on Cartesian-partitioned voxels [48], [49] to extract point cloud features for panoptic segmentation. Panoptic-PHNet [89] and GP-S3Net [90] use hybrid backbones to achieve state-of-the-art performance. Currently, most panoptic segmentation methods perform instance clustering in a bottom-up way. Some also explore the top-down style [79], [91], [92], which tends to have higher performance as shown in 2D counterparts.

TABLE 1: Categorization of LiDAR-based panoptic segmentation methods.

Seg Style	Backbone	References
Bottom-Up	Point-Based [41]	[12], [75], [76]
	Range View [5]	[77], [78], [79], [80]
	BEV [6]	[81], [82], [83]
	Cylinder [54]	[84], [85], [86], [87]
	Voxel [48]	[88]
	Hybrid	[89], [90]
Top-Down	Range View [5]	[79]
	Voxel [48]	[91], [92]

4D Panoptic Segmentation. 4D-PLS [12] extends the single frame LiDAR-based panoptic segmentation to the 4D version by constructing 4D volumes, upon which clustering-based instance segmentation is performed. Specifically, they start by selecting the point with the highest objectness score and assign points to this instance by evaluating association probabilities. 4D-StOP [76] adopts a similar strategy as 4D-PLS [12] by first constructing 4D volumes, then performing instance clustering. They use a different clustering method, where they perform DBSCAN clustering on learned geometry features. Marcuzzi *et al.* [86] use contrastive instance association on top of the single-frame panoptic segmentation framework to achieve temporally consistent instance ID assignment. Wang *et al.* [83] build their method based on efficient polar BEV to achieve real-time 4D panoptic segmentation. Our proposed 4D panoptic LiDAR segmentation method firstly constructs 4D volumes similarly to 4D-PLS [12] and 4D-StOP [76]. Different from their clustering strategies, we perform dynamic shifting on the 4D feature volumes to obtain temporally consistent thing IDs.

3 Our Approach

We structure this section into two parts, one for the single-scan version DS-Net, based on which its 4D counterpart

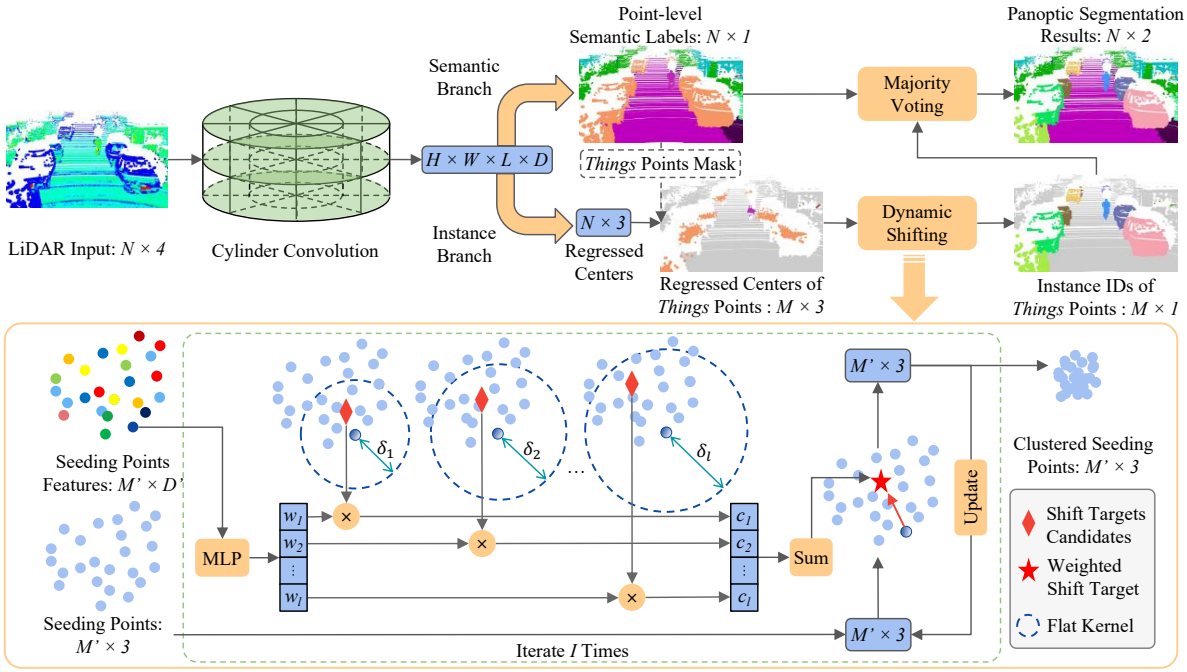


Fig. 2: **Architecture of DS-Net.** The DS-Net consists of the cylinder convolution, a semantic, and an instance branch as shown in the upper part of the figure. The regressed centers provided by the instance branch are clustered by the novel dynamic shifting module, which is shown in the bottom half.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

4D-DS-Net is further introduced. For the DS-Net part, as illustrated in Fig. 2, 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 cloud distributions (Sec. 3.2). The efficient majority voting strategy combines the semantic and instance predictions and produces panoptic segmentation results (Sec. 3.3). For the second part, we introduce a simple yet effective extension, namely 4D-DS-Net, to the task of 4D panoptic LiDAR segmentation (Sec. 3.4).

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 [15] best meets the strict requirements of high efficiency, high performance, and full mining of 3D positional relationships. The cylindrical voxel partition can produce a more even point distribution than the 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 relationships. Specifically, we use the sparse convolution to construct a U-Net [93] that operates on cylindrical voxels. The input LiDAR point clouds $P \in \mathbb{R}^{N\times 4}$ consists of N points and each point p_i has four attributes representing its XYZ coordinates (x_i, y_i, z_i) and the intensity of the corresponding reflection beams r_i. The output of the backbone is the voxel features $F_v \in \mathbb{R}^{H\times W\times L\times D}$, where D represents the dimension of the features. H, W, L are voxel resolutions and take values of $480, 360, 32$ in practice.

Semantic Branch. By applying convolution to the voxel feature from the backbone F_v, semantic logits $L_s \in \mathbb{R}^{H\times W\times L\times C}$, where C is the number of all classes and H, W, L represent voxel dimensions, are predicted for each voxel, which is then followed by a softmax operation to compute the predicted semantic label for each voxel. Point-level semantic predictions are obtained by copying voxel labels to the points inside the voxels. Considering the category imbalance in the autonomous driving scene, we choose the weighted cross-entropy loss and Lovasz loss [94] as the loss function for the semantic segmentation 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 $C_{gt} \in \mathbb{R}^{M\times 3}$, where M represents number of *things* points predicted by semantic segmentation. 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

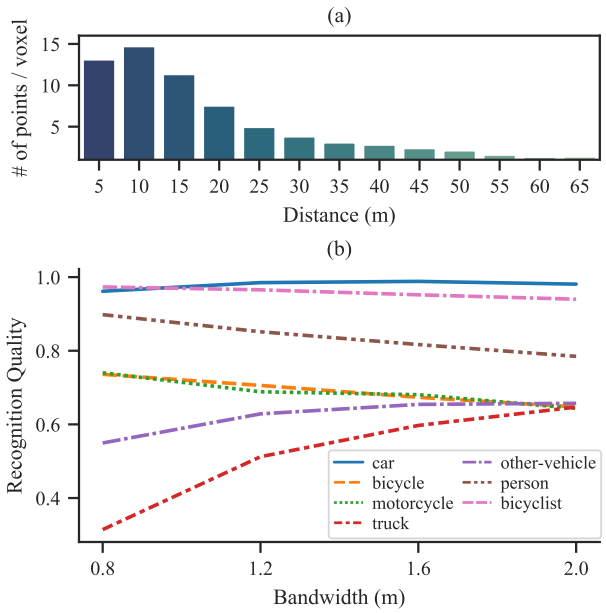


Fig. 3: (a) counts the average number of regressed centers inside each valid voxel of instances at different distances. (b) shows the effect of Different Mean Shift Bandwidth on the Recognition Quality of Different Classes.

centers $O + P$ are further clustered to obtain different instances, and then the instance IDs are assigned to them. It 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 sampled from reconstructed meshes, the LiDAR point clouds have distributions that are not suitable for normal clustering solutions used by indoor instance segmentation methods. The varying instance sizes, the sparsity, and the 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-like" distribution as shown 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 because the point cloud sparsity depends on the distance to the sensors. 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 cloud instance segmentation methods are analyzed below.

- **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, a large density difference between clusters means that the *fixed radius* cannot properly adapt to different clusters. A small radius tends to over-segment distant instances while a large radius tends to under-segment near instances.
- **DBSCAN [95] and HDBSCAN [96].** 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 proven to be effective for clustering indoor point clouds [13], [97]. 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 with LiDAR points.
- **Mean Shift [98].** The advantage of Mean Shift, which is used by [99] 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 a 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 not the ideal choice for this task.

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* operation. Inspired by [100], 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:

$$X \leftarrow X + \eta S, \tag{2}$$

where η is a scaling factor which is set to 1 in our experiments. 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 a flat kernel can be thought of as placing a query ball of a 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. The optimal bandwidth for each seeding point has to be inferred dynamically, in order for the kernel function to be adapted to instances with different sizes. A natural solution is to directly regress

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

bandwidth for each seeding point, which however is not differentiable if used with the flat kernel. Even though the 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 follows:

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

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 been gathered to the cluster centers. A simple heuristic clustering algorithm is performed to cluster the gathered 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 the 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 to shift toward their cluster centers that 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 i-th iteration of dynamic shifting is defined by the Manhattan distance between the ground truth centers C'_{gt} and the i-th dynamically calculated shift targets X_i, which can be formulated as follows:

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

Adding up all the losses of I iterations gives us the loss function L_{ds} for the dynamic shifting module: $L_{ds} = \sum_{i=1}^{I} w_i l_i$, where w_i are weights for losses of different iterations and are all set to 1 in our experiments.

3.3 Fusion of Semantic and Instance Segmentation

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 that 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 is *majority voting*. For each proposed instance, we directly assign the most frequent semantic label inside to all the points of this instance to ensure the agreement between semantic and instance segmentation results. This simple fusion strategy is not only efficient but could also revise and unify semantic predictions using instance information.

3.4 4D Panoptic LiDAR Segmentation

Based on the above proposed single version of the LiDAR-based panoptic segmentation method DS-Net, we further extend it to the task of 4D panoptic LiDAR segmentation.

To extend from single-frame panoptic segmentation to its 4D counterpart, the things IDs need to be consistent across frames. In other words, for the same instance observed in multiple frames, the target is to assign the same IDs for them. The trivial way is to append a tracking module to the instance segmentation branch to associate the predicted instance segments from previous and current frames. However, such naïve stacking of modules will inevitably lead to the compromised performance of tracking due to its dependence on the segmentation quality. Moreover, for the tracking module, it is hard to fully utilize the information provided by the consecutive LiDAR scans since it only processes the cropped partial observations. It is challenging for the tracking module to extract distinctive features from incomplete, sparse point clouds. To fully utilize the temporal information from consecutive LiDAR scans, following 4D-PLS [12], we propose to perform instance clustering in a temporally unified way, which is illustrated below.

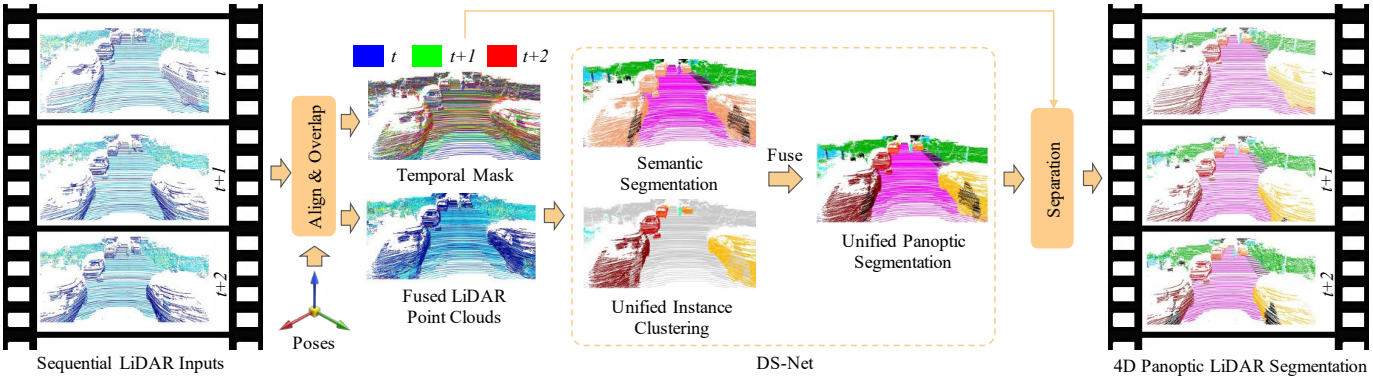


Fig. 4: **Architecture of the 4D version DS-Net (namely 4D-DS-Net).** The 4D-DS-Net takes the aligned and overlapped LiDAR scans as input. Then perform unified instance clustering to generate temporally consistent things IDs. Finally, the overlapped panoptic segmentation results are separated according to the original temporal masks.

Temporally Unified Instance Clustering. To ensure the consistency of the things IDs, we propose to use the temporally unified instance clustering to replace the explicit association. The target of such a clustering strategy is to jointly cluster all the points of the same instance from several frames to a single cluster. Then we could naturally separate these points into different frames and assign them to the same instance ID. To fit such a clustering strategy into the bottom-up pipeline, we need to modify the targets of the center regression step and the following clustering module. In the single-frame version of the pipeline, the point-level features are used to regress the center of the instances. However, in the multi-frame scenario, positions of instances, *e.g.*, cars and pedestrians, would change across frames. Therefore, if we still follow the center regression target of the single frame version, there is a great possibility of the same instance being clustered into multiple clusters because the regressed centers are too far apart due to high moving speed. To avoid such a problem, for the same instance, we propose to regress to the center of the overlapped point clouds from multiple frames, which can be formulated as:

$$C_{gt}(p_{id}) = \text{center}(\{p|p \in gt_t(id) \dots gt_{t+i}(id)\}), \quad (5)$$

where p_{id} is the point that has the things ID of id, $gt_{t+i}(id)$ represents the set of points that have the things ID of id and in frame $t+i$. After the adjusted center regression, the following clustering step is performed on the overlapped regressed centers. Unlike the 4D Volume Clustering proposed by [12], our clustering process does not take the frame timestamp of each point into consideration, which means our method is frame agnostic. Ideally, all the points of the same instance from several frames are clustered into a single cluster, which matches the target of the proposed temporally unified instance clustering. To integrate the clustering strategy into the single frame version DS-Net, we need to obtain the point-level features of several consecutive LiDAR frames from the backbone. There are two possible ways to achieve that. The first one is to merge consecutive LiDAR scans at the data level. The second one is to merge the feature map of each individual frame right after the backbone. From the perspective of computational efficiency, the first one is more efficient. After downsampling and voxelization, processing multiple frames is equivalent to the single-frame input for the backbone feature extraction part. To justify that, we have tested the GPU memory usage for both strategies given two consecutive frames are merged (examples are from the typical SemanticKITTI LiDAR scans). The data-level fusion consumes around 5483 MB of memory, while the feature-level fusion requires around 9966 MB. As for the performance, we find out that the first one is also a better strategy through extensive experiments. Therefore, based on the above analysis, we propose the 4D extension version of DS-Net, which is illustrated below.

4D Extension of DS-Net. The 4D extension version of DS-Net (namely 4D-DS-Net) for the 4D panoptic segmentation is shown in Fig. 4. Using the ego-poses estimated by SLAM algorithms [17], we align consecutive LiDAR point clouds and overlap them to get the temporally fused LiDAR point clouds. The fused LiDAR point clouds from frame t to $t+i$ are defined as:

$$P_{t:t+i} = \{p|p \in P'_t \dots P'_{t+i}\}, \quad (6)$$

$$\text{where } P'_{t+i} = ((P_{t+i}R^{-1}_{t+i} + T_{t+i}) - T_t)R_t \,. \quad (7)$$

R_{t+i} and T_{t+i} represent the rotation matrix and translation vector of frame $t+i$. The semantic segmentation branch predicts semantic labels for each point as that of the single version. The instance segmentation branch produces temporally consistent IDs for each point, which is achieved by the temporally unified instance clustering proposed above. Specifically, the foreground points are first regressed to the centers of the overlapped instances. Then, the regressed centers are further clustered by the proposed dynamic shifting network in the frame agnostic way. Such a unified instance clustering step naturally associates the same instance across frames and saves the effort of tracking algorithms. Once we ensure the consistency of things IDs in two consecutive frames, the instance IDs can be propagated to the whole sequence through overlapping frames. Specifically, in the case of using two consecutive frames as a 4D volume, there is one overlapping frame between two consecutive 4D volumes. The overlapping frame would have two sets of predicted instances, upon which we calculate the IoU as the association score. For an association score over 0.5, the two instances make a match. In this way, the instance IDs

are propagated to the whole sequence, which gives the final 4D panoptic segmentation results.

4 Experiments

We conduct experiments on two large-scale datasets: SemanticKITTI [17] and Panoptic nuScenes [18], [19]. In addition, we evaluate our extension of 4D panoptic segmentation on SemanticKITTI.

SemanticKITTI. The SemanticKITTI dataset [11] is the first dataset that presents the challenge of LiDAR-based panoptic segmentation and provides the benchmark. 4D-PLS [12] further extends the benchmark with the novel task of 4D panoptic segmentation. SemanticKITTI contains 23,201 frames for training and 20,351 frames for testing. There are 28 annotated semantic classes that 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 a temporally consistent instance ID which will be set to 0 if the point belongs to *stuff* classes.

Panoptic nuScenes. The Panoptic nuScenes dataset [19] is a large-scale LiDAR-based panoptic segmentation dataset built on nuScenes [18]. 1000 scenes with 32 semantic classes and 300k instances are provided in this dataset.

Evaluation Metrics of LiDAR-based Panoptic Segmentation. As defined in [11], the evaluation metrics of LiDAR-based panoptic segmentation are the same as that of image panoptic segmentation defined in [9] including Panoptic Quality (PQ), Segmentation Quality (SQ) and Recognition Quality (RQ) are calculated across all classes. For each class, the PQ, SQ, and RQ are defined as follows:

$$\text{PQ} = \underbrace{\frac{\sum_{(i,j)\in TP} \text{IoU}(i,j)}{|TP|}}_{\text{SQ}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{RQ}}. \quad (8)$$

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}. $\text{PQ}^{\dagger}$ 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.

Evaluation Metrics of 4D Panoptic LiDAR Segmentation. Several metrics are proposed by previous video panoptic segmentation works [10], [12], [37]. Among them, we choose to use LSTQ (LiDAR Segmentation and Tracking Quality) [12] as the evaluation metrics for 4D Panoptic Segmentation, which is defined as follows:

$$\text{LSTQ} = \sqrt{\underbrace{\frac{1}{C}\sum_{c=1}^{C} \text{IoU}(c)}_{\text{S}_{\text{cls}}} \times \underbrace{\frac{1}{T}\sum_{t\in T} \frac{\sum_{s\in S} \text{TPA}(s,t)\text{IoU}(s,t)}{|gt_{id}(t)|}}_{\text{S}_{\text{assoc}}}}, \quad (9)$$

where S_{cls} and S_{assoc} reflects the segmentation and tracking quality respectively. TPA (True Positive Association) is defined as $TPA(i,j) = |pr(i) \cap gt(j)|$, which represents the number of intersections between points that are predicted as i and the ground truth points that have the id of j.

Implementation Details of the Backbone. Following PCSeg [106] and MMDetection3D [107], for both datasets, each input point is represented as a four-dimensional vector including XYZ coordinates and intensity. The backbone voxelizes a single frame to $480 \times 360 \times 32$ voxels under the cylindrical coordinate system. For the ground truth instance center, we do not use the center of the annotated 3D bounding boxes. Instead, it is approximated by the center of its tight box that parallels axes which makes a better approximation than the mass centers of the incomplete point clouds.

Implementation Details of Dynamic Shifting. Bandwidth candidates are set to 0.2, 1.7, and 3.2 for both datasets. The number of Iterations is set to 4 for both datasets. We train the network with a learning rate of 0.002, an epoch number of 50, and a batch size of 4 on four Geforce GTX 1080Ti. The dynamic shifting module only takes 3-5 hours to train on top of a pretrained backbone.

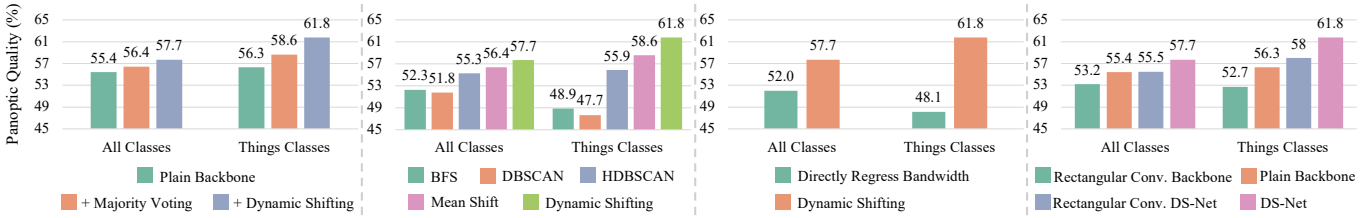
Implementation Details of 4D-DS-Net. Two consequent LiDAR scans are aligned and overlapped for the training and inference of 4D panoptic segmentation. The number of FPS downsampled points in the dynamic shifting module is set to 20000. Other hyper-parameters are the same as their single-version counterpart. For both DS-Net and 4D-DS-Net, we first pre-train the backbone with the semantic segmentation loss, then fine-tune it with the instance branch added. Finally, the dynamic shifting module is trained with previous networks fixed.

4.1 Ablation Study

Ablation on Overall Framework. To study the effectiveness of the proposed modules, we sequentially add the majority voting module and dynamic shifting module to the plain panoptic segmentation backbone. The plain backbone consists of the cylindrical backbone, semantic and instance branches, and Mean Shift as the clustering algorithm. The corresponding PQ and PQ^{Th} are reported in Fig. 5 (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 is indicated by PQ^{Th} where the DS-Net outperforms the backbone (with fusion module) by 3.2% in the 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. 5 (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 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 Fig. 5 (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 regressed bandwidth. Therefore, it is easier for the network to choose from and combine several bandwidth candidates.



(a) Ablation Study of the Overall Framework (b) Ablation Study of Clustering Algorithm (c) Ablation Study of Bandwidth Learning Style (d) Ablation Study of Backbone Choice

Fig. 5: **Ablation Study on the Validation Set of SemanticKITTI**. The proposed two modules both contribute 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.

TABLE 2: LiDAR-based panoptic segmentation results on the validation set of SemanticKITTI. All scores are in [%].

Method (Year)	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}	mIoU
KPConv [41] + PV-RCNN [2]	51.7	57.4	63.1	78.9	46.8	56.8	81.5	55.2	67.8	77.1	63.1
Cylinder3D [15] + PV-RCNN [2]	51.9	57.5	63.8	74.2	48.5	59.5	70.2	54.3	66.9	77.1	62.9
LPASD [IROS'20] [77]	36.5	46.1	-	-	-	28.2	-	-	-	-	50.7
Panoptic-TrackNet [arXiv'20] [78]	40.0	-	48.3	73.0	29.9	33.6	76.8	47.4	59.1	70.3	53.8
PointGroup [CVPR'20] [14]	46.1	54.0	56.6	74.6	47.7	55.9	73.8	45.0	57.1	75.1	55.7
PanosterK [RA-L'21] [75]	55.6	-	66.8	79.9	56.6	65.8	-	-	-	-	61.1
Panoptic-PolarNet [CVPR'21] [81]	59.1	64.1	70.2	78.3	65.7	74.7	87.4	54.3	66.9	71.6	64.5
EfficientLPS [TRO'21] [79]	59.2	65.1	69.8	75.0	58.0	68.2	78.0	**60.9**	71.0	72.8	64.9
GP-S3Net [ICCV'21] [90]	63.3	71.5	75.9	**81.4**	70.2	80.1	86.2	58.3	72.9	77.9	**73.0**
Location-Guided [TIV'22] [82]	59.0	63.1	69.4	78.7	65.3	73.5	**88.5**	53.9	66.4	71.6	61.4
Panoptic-PHNet [CVPR'22] [89]	61.7	-	-	-	69.3	-	-	-	-	-	65.7
PUPS [AAAI'23] [101]	**64.4**	**68.6**	**81.5**	74.1	**73.0**	**92.6**	79.3	58.1	**73.5**	70.4	-
DS-Net (Ours)	57.7	63.4	68.0	77.6	61.8	68.8	78.2	54.8	67.3	77.1	63.5
DS-Net* (Ours)	61.4	65.2	72.7	79.0	65.2	72.3	79.3	57.9	71.1	**79.3**	69.6

Ablation on Backbone Choice. To demonstrate that the dynamic shifting module can apply to different backbones, we report the performance of a rectangular convolution version of plain backbone and DS-Net Fig. 5 (d). An improvement of 2.3% in terms of PQ on both the rectangular convolution version and cylinder convolution version of the plain backbone is achieved on the validation set of SemanticKITTI, which shows the proposed dynamic shifting module can also work with other backbones.

4.2 Comparisons on SemanticKITTI

In Tab. 2 and Tab. 3, we summarize all available results on SemanticKITTI and annotate them with venue and year for a clear and thorough comparison. DS-Net* reports the result of changing the backbone from Cylinder3D to SPVCNN [56]. Results show that the original DS-Net achieves superior performances among concurrent works. After migrating to a stronger backbone, DS-Net still achieves competitive scores in both validation and test splits, showing the effectiveness of the proposed dynamic shifting network. 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. Moreover, methods that use hybrid backbones *e.g.*, GP-S3Net and Panoptic-PHNet, have better overall performance than single-backbone ones. This observation may serve as strong clues for future researchers when choosing backbones. This also clearly indicates that hybrid backbone structure design is also a promising future direction.

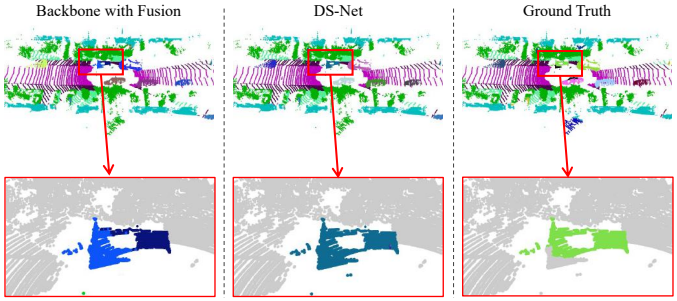


Fig. 6: Qualitative Results on SemanticKITTI. The dynamic shifting module helps to correctly segment instances with complex shapes and densities.

We show some visualizations of our results in Fig. 6. The left shows the results of the bare backbone with the majority voting module. The middle one shows the results of the DS-Net and the right one is the ground truth. Our DS-Net is capable of correctly handling instances with complex shapes and densities, while the backbone method tends to either over- or under-segment in these cases. For more visualization, please refer to the supplementary material.

4.3 Comparisons on Panoptic nuScenes

We report results on validation and test sets of Panoptic nuScenes in Tab. 4 and Tab. 5. We collect all the methods that report results on this dataset and annotate them with venues and years for a complete comparison. Similarly, DS-Net*

TABLE 3: LiDAR-based panoptic segmentation results on the test set of SemanticKITTI. All scores are in [%].

Method (Year)	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}	mIoU
KPConv [41] + PointPillars [1]	44.5	52.5	54.4	80.0	32.7	38.7	81.5	53.1	65.9	79.0	58.8
RangeNet++ [4] + PointPillars [1]	37.1	45.9	47.0	75.9	20.2	25.2	75.2	49.3	62.8	76.5	52.4
KPConv [41] + PV-RCNN [2]	50.2	57.5	61.4	80.0	43.2	51.4	80.2	55.9	68.7	79.9	62.8
Cylinder3D [54] + SLR [102]	56.0	62.6	67.4	82.1	51.8	61.0	84.2	59.1	72.1	80.6	67.9
LPASD [IROS'20] [77]	38.0	47.0	48.2	76.5	25.6	31.8	76.8	47.1	60.1	76.2	50.9
Panoptic-TrackNet [arXiv'20] [78]	43.1	50.7	53.9	78.8	28.6	35.5	80.4	53.6	67.3	77.7	52.6
4D-PLS [CVPR'21] [12]	50.3	57.8	61.0	81.6	-	-	-	-	-	-	61.3
PanosterK [RA-L'21] [75]	52.7	59.9	64.1	80.7	49.4	58.5	83.3	55.1	68.2	78.8	59.9
Panoptic-PolarNet [CVPR'21] [81]	54.1	60.7	65.0	81.4	53.3	60.6	87.2	54.8	68.1	77.2	59.5
PC-Cluster [ICRA'22] [85]	56.5	63.1	67.9	82.3	52.9	62.1	84.8	59.1	72.1	80.6	68.2
CPSeg [arXiv'21] [103]	57.0	63.5	68.8	82.2	55.1	64.1	86.1	58.4	72.3	79.3	62.7
EfficientLPS [TRO'21] [79]	57.4	63.2	68.7	83.0	53.1	60.5	87.8	60.5	74.6	79.5	61.4
GP-S3Net [ICCV'21] [90]	60.0	69.0	72.1	82.0	65.0	74.5	86.6	56.4	70.4	78.7	**70.8**
SSDF [RS'22] [83]	54.6	61.5	65.5	81.7	54.0	61.9	86.7	55.1	68.2	78.1	60.6
Location-Guided [TIV'22] [82]	54.7	61.1	65.5	81.6	54.6	62.1	87.3	54.8	68.0	77.5	59.0
SMAC-Seg [ICRA'22] [80]	56.1	62.5	67.9	82.0	53.0	61.8	85.6	58.4	72.3	79.3	63.3
PVCL [ICRA'22] [87]	59.1	65.7	69.6	84.0	59.8	66.7	89.2	58.6	71.6	80.3	64.0
SCAN [arXiv'22] [88]	61.5	67.5	84.5	72.1	61.4	88.1	69.3	**61.5**	**81.8**	74.1	67.7
Panoptic-PHNet [CVPR'22] [89]	61.5	**67.9**	72.1	**84.8**	63.8	70.4	**90.7**	59.9	73.3	80.5	66.0
PUPS [AAAI'23] [101]	**62.2**	65.8	**84.2**	72.8	**65.7**	**90.6**	72.7	59.6	79.5	73.1	-
DS-Net (Ours)	55.9	62.5	66.7	82.3	55.1	62.8	87.2	56.5	69.5	78.7	61.6
DS-Net* (Ours)	59.8	66.5	70.6	83.9	59.1	66.7	88.2	60.3	73.4	**80.7**	67.7

TABLE 4: LiDAR-based panoptic segmentation results on the validation set of Panoptic nuScenes. All scores are in [%].

Method (Year)	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}	mIoU
Panoptic-TrackNet [arXiv'20] [78]	51.4	56.2	63.3	80.2	45.8	55.9	81.4	60.4	75.5	78.3	58.0
VIN [arXiv'21] [91]	51.7	57.4	61.8	82.6	45.7	53.7	83.6	61.8	75.4	80.9	73.7
EfficientLPS [TRO'21] [79]	59.2	62.8	82.9	70.7	51.8	80.6	62.7	71.5	84.3	84.1	69.4
Panoptic-PolarNet [CVPR'21] [81]	63.4	67.2	75.3	83.9	59.2	70.3	84.1	70.4	83.5	83.6	66.9
CPSeg [arXiv'21] [103]	68.9	73.6	80.4	84.9	68.7	78.4	86.8	69.2	82.5	83.1	72.7
PVCL [ICRA'22] [87]	64.9	67.8	77.9	81.6	59.2	72.5	79.7	67.6	79.1	77.3	73.9
SCAN [arXiv'22] [88]	65.1	68.9	85.7	75.3	60.6	85.7	70.2	72.5	85.7	83.8	77.4
SMAC-Seg [ICRA'22] [80]	67.0	71.8	78.2	85.0	65.2	74.2	87.1	68.8	82.2	82.9	72.2
Panoptic-PHNet [CVPR'22] [89]	74.7	**77.7**	84.2	**88.2**	74.0	82.5	**89.0**	**75.9**	**86.9**	**86.8**	79.7
LidarMultiNet [arXiv'22] [92]	**81.8**	-	-	-	-	-	-	-	-	-	**83.6**
PUPS [AAAI'23] [101]	74.7	77.3	**89.4**	83.3	**75.4**	**91.8**	81.9	73.6	85.3	85.6	-
DS-Net (Ours)	60.6	63.7	72.6	81.2	53.3	65.2	79.2	72.7	85.0	84.7	73.7
DS-Net* (Ours)	64.7	67.6	76.1	83.5	58.6	64.2	82.8	74.7	86.5	85.5	76.3

reports the results of changing the backbone to SPVCNN [56]. Among concurrent works, our method achieves the best results. Admittedly, there still exists a gap between our results and more recent solutions. The first three lines in Tab. 5 are combinations of strong segmentation and detection methods. Panoptic-PHNet and LidarMultiNet even surpass 80% in the PQ score. These methods are either top-down methods or heavily borrow structures from detection solutions, *e.g.*, center heatmap. Their strong performances show that for datasets like panoptic nuScenes, where LiDAR scans are sparser, pure clustering-based methods have more disadvantages than top-down ones.

4.4 4D Panoptic LiDAR Segmentation Results

Comparison Methods. Since the task is fairly new, we choose to compare with the first work [12] that proposes this task, denoted as '4D-PLS', two recent works 4D-StOP [76], CIA [86] and several 'Semantic Segmentation + 3D Object Detection + Tracking' assembled baseline methods. Besides, we also construct a baseline method 'DS-Net + Tracking' by appending a tracking module [10] to the instance segmentation branch. As discussed in Sec. 3.4, we also implement the feature map fusion on top of DS-Net, namely 'DS-Net + Feat. Fus.'. Specifically, we perform a max pooling operation on the aligned 3D feature maps extracted from consecutive LiDAR frames by the backbone. Then the fused feature map is fed to semantic and instance branches. We refer to our proposed method as '4D-DS-Net'. 4D-DS-Net* indicates changing the backbone to Cylinder3D++ [54].

Evaluation Results. As shown in Tab. 6 and Tab. 7, our proposed method surpasses all assembled baseline methods and the state-of-the-art methods in terms of the main metric LSTQ in both validation and test sets. '4D-DS-Net' surpasses 'DS-Net + Tracking' by 2.1% in terms of LSTQ on the validations set, which proves that simply stacking modules is hard to fully utilize the temporal information as mentioned in Sec. 3.4. Note that "DS-Net + Feat. Fus." has a better association score S_{assoc} but worse segmentation score S_{cls} than 4D-DS-Net. We think there might be two reasons. **a)** The feature-level fusion extracts features from

TABLE 5: LiDAR-based panoptic segmentation results on the test set of Panoptic nuScenes. All scores are in [%].

Method (Year)	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}	mIoU
SPVNAS [56] + CenterPoint [104]	72.2	76.0	81.2	88.5	71.7	79.4	89.7	73.2	84.2	86.4	76.9
Cylinder3D++ [54] + CenterPoint [104]	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 [105] + CenterPoint [104]	76.8	80.6	85.4	89.5	79.8	86.8	91.8	71.8	83.0	85.7	78.8
EfficientLPS [TRO'21] [79]	62.4	66.0	74.1	83.7	57.2	68.2	83.6	71.1	84.0	83.8	66.7
Panoptic-PolarNet [CVPR'21] [81]	63.6	67.1	75.1	84.3	59.0	69.8	84.3	71.3	83.9	84.2	67.0
Panoptic-PHNet [CVPR'22] [89]	80.1	**82.8**	**87.6**	**91.1**	**82.1**	**88.1**	**93.0**	**76.6**	**86.6**	**87.9**	80.2
LidarMultiNet [arXiv'22] [92]	**81.4**	-	-	-	-	-	-	-	-	-	**84.0**
DS-Net (Ours)	59.2	62.6	68.8	83.8	51.1	59.6	82.6	72.6	84.0	85.8	75.1
DS-Net* (Ours)	65.8	68.9	75.2	85.9	60.7	69.6	85.7	74.3	85.2	86.2	77.1

TABLE 6: 4D panoptic LiDAR segmentation results on the validation set of SemanticKITTI. All scores are reported in [%]. RN: RangeNet++ [4]; KP: KPConv [41]; PP: PointPillars [1]; MOT: Multi-Object Tracking [108]; SFP: Scene Flow based Propagation [109].

Method	LSTQ	S_{assoc}	S_{cls}	IoU^{st}	IoU^{th}
RN + PP + MOT	43.8	36.3	52.8	60.5	42.2
KP + PP + MOT	46.3	37.6	57.0	64.2	54.1
RN + PP + SFP	43.4	35.7	52.8	60.5	42.2
KP + PP + SFP	46.0	37.1	57.0	64.2	54.1
MOPT [78]	24.8	11.7	52.4	62.4	45.3
4D-PLS (2-scan) [12]	59.9	58.8	61.0	65.0	63.1
4D-PLS (4-scan) [12]	62.7	65.1	60.5	**65.4**	61.3
4D-StOP (2-scan) [76]	66.4	71.8	61.4	64.9	64.1
4D-StOP (4-scan) [76]	67.0	**74.4**	60.3	65.3	60.9
DS-Net + Tracking	65.9	68.4	63.1	64.0	61.9
DS-Net + Feat. Fus.	67.8	72.1	63.7	64.2	63.1
4D-DS-Net (2-scan)	68.0	71.3	64.8	64.5	65.3
4D-DS-Net (3-scan)	**68.3**	71.5	**65.1**	64.4	**66.0**
4D-DS-Net (4-scan)	68.1	71.3	64.9	64.4	65.4

TABLE 7: 4D panoptic LiDAR segmentation results on the test set of SemanticKITTI. All scores are reported in [%]. RN: RangeNet++ [4]; KP: KPConv [41]; PP: PointPillars [1]; MOT: Multi-Object Tracking [108]; SFP: Scene Flow based Propagation [109].

Method	LSTQ	S_{assoc}	S_{cls}	IoU^{st}	IoU^{th}
RN + PP + MOT	35.5	24.1	52.4	64.5	35.8
KP + PP + MOT	38.0	25.9	55.9	66.9	47.7
RN + PP + SFP	34.9	23.3	52.4	64.5	35.8
KP + PP + SFP	38.5	26.6	55.9	66.9	47.7
4D-PLS [12]	56.9	56.4	57.4	66.9	51.6
CIA [86]	63.1	65.7	60.6	66.9	52.0
4D-StOP (2-scan) [76]	62.9	67.3	58.8	**68.3**	53.3
4D-StOP (4-scan) [76]	63.9	**69.5**	58.8	67.7	**53.8**
4D-DS-Net (2-scan)	63.7	67.6	60.0	66.2	51.6
4D-DS-Net (3-scan)	**64.2**	67.5	**61.0**	66.8	52.9
4D-DS-Net (4-scan)	64.0	67.4	60.6	66.9	52.3

each LiDAR scan individually, making the instance-level feature extraction less affected by the "trailing" issue as discussed in Q1. This leads to better 4D instance clustering, which favors the association score S_{assoc}. **b)** 4D-DS-Net takes the data-level fusion strategy, meaning several consecutive LiDAR frames are aligned, overlapped, and sent to the feature extraction backbone as a single scan. Originally partially observed *things* and *stuff* now have information from multiple viewing angles, which lowers the difficulty of semantic segmentation. Therefore, the data-level fusion strategy favors the segmentation score S_{cls}. We have also ablated on the number of frames used for 4D panoptic segmentation on both validation and test set of SemanticKITTI [12], [17], as shown in Tab. 6 and 7. It is interesting to see that *"3-scan"* version achieves the best result among the three variations. We think this is because, with more scans overlapped, the "trailing" issue poses more challenges to instance clustering in the 4D volume. However, more scans would give the network denser point clouds and more observations of the scene, which favors the semantic understanding. Therefore, *"3-scan"* is the balance point between these two factors. We have also included this discussion in the main manuscript. However, the amount of memory and computation overhead of 'DS-Net + Feat. Fus.' compared to that of '4D-DS-Net' still justifies our preference for data-level fusion. For qualitative evaluations, please refer to the supplementary material.

TABLE 8: Ablation results of Single Frame PQ Evaluation of 4D-DS-Net on the validation set of SemanticKITTI.

Name	PQ	PQ†	PQ^{Th}	PQ^{St}	mIoU
DS-Net	57.7	63.4	61.8	54.8	63.5
DS-Net + Feat. Fus.	58.6	63.9	63.4	55.0	63.7
4D-DS-Net	**59.5**	**64.5**	**64.4**	**55.9**	**64.8**

4D Panoptic Segmentation Improves the Single Frame PQ Evaluation. We also evaluate the single frame metrics using the 4D version of DS-Net on the validation set of SemanticKITTI. As shown in Tab. 8, the 4D version of DS-Net surpasses the single frame DS-Net by 1.8% in terms of PQ. It shows that the temporal information can largely enrich the semantic information extracted by the backbone and therefore improve the overall performance. The improved single-frame segmentation quality also explains the better performance in the task of 4D panoptic LiDAR segmentation. Moreover, '4D-DS-Net' also outperforms 'DS-Net + Feat. Fus.' by 0.9% in terms of PQ, which shows the superiority of data-level fusion over simple feature-level fusion. Of course, more complex feature fusion could be designed and has the potential to outperform data-level fusion. But with minimal memory and computation overhead, data-level fusion is the first choice here.

4.5 Further Analysis

Robust to Parameter Settings. As shown in Tab. 9, six sets of bandwidth candidates are set for independent training, and the corresponding results are reported. The stable results

TABLE 9: Ablation results of different bandwidth candidates settings. All scores are reported 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

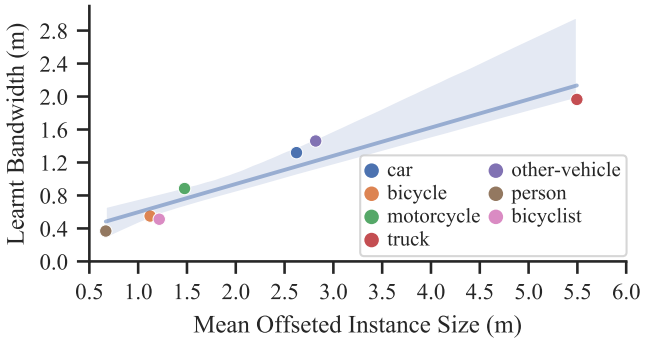


Fig. 7: **Proportional Relationship Between Sizes and the Learned Bandwidths.** The x-axis represents the class-wise average size of the regressed center of instances while the y-axis stands for the average learned bandwidth of different *things* classes.

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 some hyperparameter searching, DS-Net can automatically adjust to different instance sizes and point distributions and maintain stable clustering quality.

Interpretable Learned Bandwidths. By averaging the bandwidth candidates weighted by the learned weights, the learned bandwidth for every point could be approximated accordingly. The average learned bandwidths of different classes are shown in Fig. 7. As can be seen, the average learned bandwidths are roughly proportional to the instance sizes of the corresponding classes, which is consistent with the expectation that dynamic shifting can dynamically adjust to different instance sizes.

Visualization of Dynamic Shifting Iterations. As visualized in Fig. 8, 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 represent higher learned bandwidth and bluer points represent 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 converged around the instance centers.

Learned Bandwidths of Different Iterations. The average learned bandwidths of different iterations are shown in Fig. 9. As expected, as the iteration rounds grow, points of the same instance gather tighter which usually requires smaller bandwidths. After four iterations, learned bandwidths of most classes have dropped to 0.2, which is the lowest they can get, meaning that four iterations are enough for *things* points to converge to cluster centers, which further validates the conclusion made in the last paragraph.

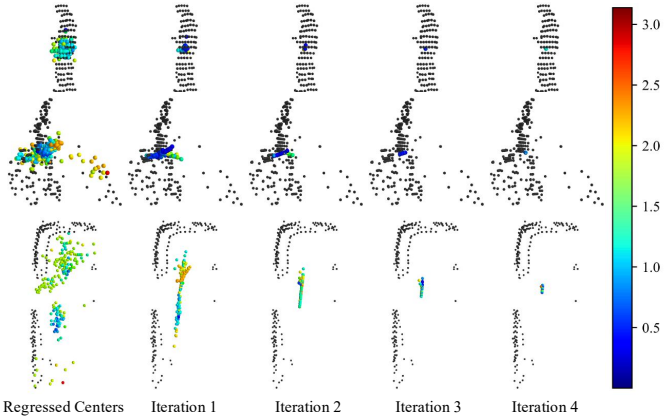


Fig. 8: **Visualization of Dynamic Shifting Iterations.** The black points are the original LiDAR point clouds of instances. The colored points are seeding points. From left to right, with the iteration number increases, the seeding points converge to cluster centers.

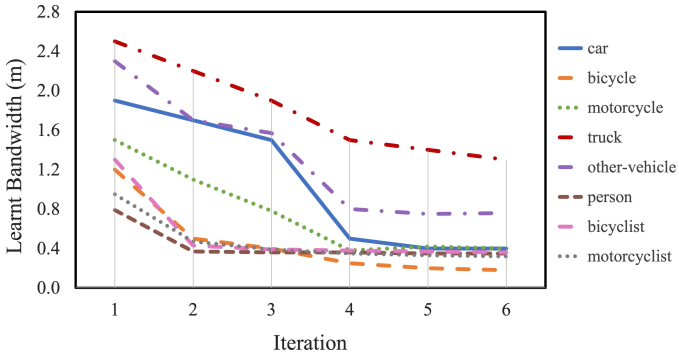


Fig. 9: **Relationship between Learned Bandwidths and Iterations.** With the number of iterations increasing, the learned bandwidth decreases. As can be seen from the curves, at the fifth iteration, the learned bandwidths of most classes drop near the lower limit.

Inference Time Analysis. The inference time of different modules of DS-Net on nuScenes is reported in Tab. 10. The dynamic shifting module takes 33.1 ms per frame, which is attributed to downsampling and GPU-accelerated matrix operations. After kernel operations, the seeding points are mostly converged. Therefore, the time of the final cluster step is negligible. Compared to the Mean Shift, which takes 190.3 ms, the proposed dynamic shifting module is efficient and performs better.

Failure Cases. The fast-moving instances would lead to overlapped points with long trailing, which increases difficulty for instance clustering in 4D volumes. This would result in the over-segmentation in the 4D volumes, which leads to wrongly changing the instance ID during tracking. Other than the trailing issue, hard classes are prone to be wrongly segmented. For qualitative evaluation, please kindly refer to the supplementary material.

TABLE 10: Inference time breakdown comparisons between our Dynamic Shift (DS) and the Mean Shift (MS) counterpart. The numbers reported are in [ms].

Module	Voxelize	Cylinder	Sem	Ins	Cluster	Fusion	Other	All
MS	35.3	106.7	5.3	44.4	190.3	3.3	36.7	422.0
DS	35.3	106.7	5.3	44.4	33.1	3.3	36.7	264.8

5 Conclusion

With the goal of providing a holistic perception of autonomous driving, we proposed a unified 3D and 4D LiDAR-based panoptic segmentation framework. In order to tackle the challenge brought by the non-uniform distributions of LiDAR point clouds, we proposed the novel DS-Net which is specifically designed for effective panoptic segmentation of LiDAR point clouds. The novel dynamic shifting module adaptively shifts regressed centers of instances with different densities and varying sizes. By constructing 4D data volumes and performing dynamic shifting clustering on them, we naturally extend the single-frame version of DS-Net to the 4D version. Through extensive experiments on two large-scale datasets, we show the competitive performance of DS-Net and 4D-DS-Net. 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 study is supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOE-T2EP20221-0012), NTU NAP, and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). 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] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[2] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[3] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[4] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 4213–4220.

[5] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 1887–1893.

[6] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9601–9610.

[7] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu, "Segment any point cloud sequences by distilling vision foundation models," in *Advances in Neural Information Processing Systems*, 2023.

[8] R. Chen, Y. Liu, L. Kong, N. Chen, X. Zhu, Y. Ma, T. Liu, and W. Wang, "Towards label-free scene understanding by vision foundation models," in *Advances in Neural Information Processing Systems*, 2023.

[9] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.

[10] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Video panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9859–9868.

[11] J. Behley, A. Milioto, and C. Stachniss, "A benchmark for lidar-based panoptic segmentation based on kitti," *arXiv preprint arXiv:2003.02371*, 2020.

[12] M. Aygun, A. Osep, M. Weber, M. Maximov, C. Stachniss, J. Behley, and L. Leal-Taixé, "4d panoptic lidar segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5527–5537.

[13] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner, "3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9031–9040.

[14] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "Pointgroup: Dual-set point grouping for 3d instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4867–4876.

[15] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9939–9948.

[16] S. Shen, Y. Cai, W. Wang, and S. Scherer, "Dytanvo: Joint refinement of visual odometry and motion segmentation in dynamic environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4048–4055.

[17] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9297–9307.

[18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 621–11 631.

[19] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada, "Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3795–3802, 2022.

[20] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *IEEE/CVF International Conference on Computer Vision*, 2015, pp. 1377–1385.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE/CVF International Conference on Computer Vision*, 2017, pp. 2961–2969.

[22] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6399–6408.

[23] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8277–8286.

[24] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang, "Attention-guided unified network for panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7026–7035.

[25] Y. Wu, G. Zhang, Y. Gao, X. Deng, K. Gong, X. Liang, and L. Lin, "Bidirectional graph reasoning network for panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9080–9089.

[26] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang, "An end-to-end network for panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6172–6181.

[27] Y. Yang, H. Li, X. Li, Q. Zhao, J. Wu, and Z. Lin, "Sognet: Scene overlap graph network for panoptic segmentation," *arXiv preprint arXiv:1911.07527*, 2019.

[28] Y. Chen, G. Lin, S. Li, O. Bourahla, Y. Wu, F. Wang, J. Feng, M. Xu, and X. Li, "Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation," in *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2020, pp. 3793–3802.

[29] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "Upsnet: A unified panoptic segmentation network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8818–8826.

[30] Q. Li, X. Qi, and P. H. Torr, "Unifying training and inference for panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 320–13 328.

[31] H. Wang, R. Luo, M. Maire, and G. Shakhnarovich, "Pixel consensus voting for panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9464–9473.

[32] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 475–12 485.

[33] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," *arXiv preprint arXiv:2003.07853*, 2020.

[34] X. Li, W. Zhang, J. Pang, K. Chen, G. Cheng, Y. Tong, and C. C. Loy, "Video k-net: A simple, strong, and unified baseline for video segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[35] X. Li, H. Yuan, W. Zhang, J. Pang, G. Cheng, and C. C. Loy, "Tube-link: A flexible cross tube baseline for universal video segmentation," *arXiv preprint arXiv:2303.12782*, 2023.

[36] J. Yang, W. Peng, X. Li, Z. Guo, L. Chen, B. Li, Z. Ma, K. Zhou, W. Zhang, C. C. Loy, and Z. Liu, "Panoptic video scene graph generation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[37] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3997–4008.

[38] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[40] X. Liu, Z. Han, F. Hong, Y.-S. Liu, and M. Zwicker, "Lrc-net: Learning discriminative features on point clouds by encoding local region contexts," *Computer Aided Geometric Design*, vol. 79, p. 101859, 2020.

[41] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6411–6420.

[42] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions On Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[43] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[44] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.

[45] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5589–5598.

[46] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[47] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[48] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[49] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 547–12 556.

[50] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in *International Conference on Robotics and Automation*, 2019, pp. 4376–4382.

[51] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, "Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation," *arXiv preprint arXiv:2004.01803*, 2020.

[52] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu, "Rethinking range view representation for lidar segmentation," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 228–240.

[53] X. Xu, L. Kong, H. Shuai, and Q. Liu, "Frnet: Frustum-range networks for scalable lidar segmentation," *arXiv preprint arXiv:2312.04484*, 2023.

[54] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9939–9948.

[55] X. Zhu, H. Zhou, T. Wang, F. Hong, W. Li, Y. Ma, H. Li, R. Yang, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar-based perception," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[56] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European Conference on Computer Vision*, 2020, pp. 685–702.

[57] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 024–16 033.

[58] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, "Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation," *arXiv preprint arXiv:2012.04934*, 2020.

[59] Y. Liu, X. L. Runnan Chen, L. Kong, Y. Yang, Z. Xia, Y. Bai, X. Zhu, Y. Ma, Y. Li, Y. Qiao, and Y. Hou, "Uniseg: A unified multi-modal lidar segmentation network and the openpcseg codebase," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 662–21 673.

[60] L. Kong, J. Ren, L. Pan, and Z. Liu, "Lasermix for semi-supervised lidar semantic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 705–21 715.

[61] L. Kong, N. Quader, and V. E. Liong, "Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation," in *IEEE Conference on Robotics and Automation*, 2023, pp. 9338–9345.

[62] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, and W. Wang, "Clip2scene: Towards label-efficient 3d scene understanding by clip," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7020–7030.

[63] J. Ren, L. Pan, and Z. Liu, "Benchmarking and analyzing point cloud classification under corruptions," in *International Conference on Machine Learning*, 2022, pp. 18 559–18 575.

[64] J. Ren, L. Kong, L. Pan, and Z. Liu, "Pointcloud-c: Benchmarking and analyzing point cloud perception robustness under corruptions," *Preprint*, 2022.

[65] L. Kong, S. Xie, H. Hu, L. X. Ng, B. R. Cottereau, and W. T. Ooi, "Robodepth: Robust out-of-distribution depth estimation under corruptions," in *Advances in Neural Information Processing Systems*, 2023.

[66] L. Kong, Y. Niu, S. Xie, H. Hu, L. X. Ng, B. Cottereau, D. Zhao, L. Zhang, H. Wang, W. T. Ooi, R. Zhu, Z. Song, L. Liu, T. Zhang, J. Yu, M. Jing, P. Li, X. Qi, C. Jin, Y. Chen, J. Hou, J. Zhang, Z. Kan, Q. Lin, L. Peng, M. Li, D. Xu, C. Yang, Y. Yao, G. Wu, J. Kuai, X. Liu, J. Jiang, J. Huang, B. Li, J. Chen, S. Zhang, S. Ao, Z. Li, R. Chen, H. Luo, F. Zhao, and J. Yu, "The robodepth challenge: Methods and advancements towards robust depth estimation," *arXiv preprint arXiv:2307.15061*, 2023.

[67] L. Kong, Y. Liu, X. Li, R. Chen, W. Zhang, J. Ren, L. Pan, K. Chen, and Z. Liu, "Robo3d: Towards robust and reliable 3d perception against corruptions," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 994–20 006.

[68] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2569–2578.

[69] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4096–4105.

[70] Q.-H. Pham, T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "Jsis3d: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8827–8836.

[71] L. Zhao and W. Tao, "Jsnet: Joint instance and semantic segmentation of 3d point clouds." in *AAAI Conference on Artificial Intelligence*, 2020, pp. 12 951–12 958.

[72] L. Han, T. Zheng, L. Xu, and L. Fang, "Occuseg: Occupancy-aware 3d instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2940–2949.

[73] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying unknown instances for autonomous driving," in *Conference on Robot Learning*, 2019.

[74] F. Zhang, C. Guan, J. Fang, S. Bai, R. Yang, P. Torr, and V. Prisacariu, "Instance segmentation of lidar point clouds," *International Conference on Robotics and Automation*, 2020.

[75] S. Gasperini, M.-A. N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari, "Panoster: End-to-end panoptic segmentation of lidar point clouds," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3216–3223, 2021.

[76] L. Kreuzberg, I. E. Zulfikar, S. Mahadevan, F. Engelmann, and B. Leibe, "4d-stop: Panoptic segmentation of 4d lidar using spatio-temporal object proposal generation and aggregation," *arXiv preprint arXiv:2209.14858*, 2022.

[77] A. Milioto, J. Behley, C. McCool, and C. Stachniss, "Lidar panoptic segmentation for autonomous driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[78] J. V. Hurtado, R. Mohan, W. Burgard, and A. Valada, "Mopt: Multi-object panoptic tracking," *arXiv preprint arXiv:2004.08189*, 2020.

[79] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada, "Efficientlps: Efficient lidar panoptic segmentation," *IEEE Transactions on Robotics*, 2021.

[80] E. Li, R. Razani, Y. Xu, and B. Liu, "Smac-seg: Lidar panoptic segmentation via sparse multi-directional attention clustering," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 9207–9213.

[81] Z. Zhou, Y. Zhang, and H. Foroosh, "Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 194–13 203.

[82] G. Xian, C. Ji, L. Zhou, G. Chen, J. Zhang, B. Li, X. Xue, and J. Pu, "Location-guided lidar-based panoptic segmentation for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2022.

[83] W. Wang, X. You, J. Yang, M. Su, L. Zhang, Z. Yang, and Y. Kuang, "Lidar-based real-time panoptic segmentation via spatiotemporal sequential data fusion," *Remote Sensing*, vol. 14, no. 8, p. 1775, 2022.

[84] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu, "Lidar-based panoptic segmentation via dynamic shifting network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 090–13 099.

[85] Y. Zhao, X. Zhang, and X. Huang, "A divide-and-merge point cloud clustering algorithm for lidar panoptic segmentation," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 7029–7035.

[86] R. Marcuzzi, L. Nunes, L. Wiesmann, I. Vizzo, J. Behley, and C. Stachniss, "Contrastive instance association for 4d panoptic segmentation using sequences of 3d lidar scans," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1550–1557, 2022.

[87] M. Liu, Q. Zhou, H. Zhao, J. Li, Y. Du, K. Keutzer, L. Du, and S. Zhang, "Prototype-voxel contrastive learning for lidar point cloud panoptic segmentation," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 9243–9250.

[88] S. Xu, R. Wan, M. Ye, X. Zou, and T. Cao, "Sparse cross-scale attention network for efficient lidar panoptic segmentation," *arXiv preprint arXiv:2201.05972*, 2022.

[89] J. Li, X. He, Y. Wen, Y. Gao, X. Cheng, and D. Zhang, "Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 809–11 818.

[90] R. Razani, R. Cheng, E. Li, E. Taghavi, Y. Ren, and L. Bingbing, "Gp-s3net: Graph-based panoptic sparse semantic segmentation network," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 076–16 085.

[91] Y. Zhong, M. Zhu, and H. Peng, "Vin: Voxel-based implicit network for joint 3d object detection and segmentation for lidars," *arXiv preprint arXiv:2107.02980*, 2021.

[92] D. Ye, W. Chen, Z. Zhou, Y. Xie, Y. Wang, P. Wang, and H. Foroosh, "Lidarmutlinet: Unifying lidar semantic segmentation, 3d object detection, and panoptic segmentation in a single multi-task network," *arXiv preprint arXiv:2206.11428*, 2022.

[93] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[94] M. Berman, A. Rannen Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421.

[95] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.

[96] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2013, pp. 160–172.

[97] D. Zhang, J. Chun, S. K. Cha, and Y. M. Kim, "Spatial semantic embedding network: Fast 3d instance segmentation with deep metric learning," *arXiv preprint arXiv:2007.03169*, 2020.

[98] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[99] J. Lahoud, B. Ghanem, M. Pollefeys, and M. R. Oswald, "3d instance segmentation via multi-task metric learning," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9256–9266.

[100] S. Kong and C. C. Fowlkes, "Recurrent pixel embedding for instance grouping," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9018–9028.

[101] S. Su, J. Xu, H. Wang, Z. Miao, X. Zhan, D. Hao, and X. Li, "Pups: Point cloud unified panoptic segmentation," *arXiv preprint arXiv:2302.06185*, 2023.

[102] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 5067–5073.

[103] E. Li, R. Razani, Y. Xu, and B. Liu, "Cpseg: Cluster-free panoptic segmentation of 3d lidar point clouds," *arXiv preprint arXiv:2111.01723*, 2021.

[104] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 784–11 793.

[105] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "Af2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 547–12 556.

[106] Y. Liu, Y. Bai, L. Kong, R. Chen, Y. Hou, B. Shi, and Y. Li, "Pcseg: An open source point cloud segmentation codebase," https://github.com/PJLab-ADG/PCSeg, 2023.

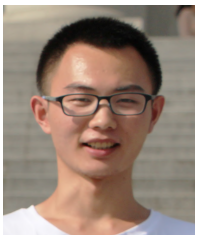
[107] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," https://github.com/open-mmlab/mmdetection3d, 2020.

[108] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 10 359–10 366.

[109] H. Mittal, B. Okorn, and D. Held, "Just go with the flow: Self-supervised scene flow estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 177–11 185.

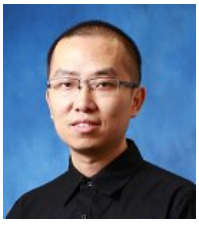
Fangzhou Hong received the BEng degree in Software Engineering from Tsinghua University, China, in 2020. He is currently a Ph.D. student in the School of Computer Science and Engineering at Nanyang Technological University. His research interests lie on the computer vision and deep learning. Particularly he is interested in 3D representation learning.

Lingdong Kong received the B.Eng. degree from the South China University of Technology, Guangzhou, China, in 2019, and the M.Sc. and M.Eng. degrees from Nanyang Technological University, Singapore, in 2020 and 2022, respectively. He is currently a Ph.D. student in the School of Computing, Department of Computer Science, National University of Singapore. His research interests include 3D perception, domain adaptation, and semi-supervised learning.



Hui Zhou received the bachelors and masters degree at university of science and electronic technology of china(UESTC) in 2015, 2018. He is currently a research scientist for autonomous driving in Sensetime Research. His research interests include computer vision and machine learning.

Xinge Zhu received the BEng degree in computer science from Shandong University, China, in 2015. He is working toward the PhD degree in The Chinese University of Hong Kong, under the supervision of Professor Dahua Lin. His research interests lie on the computer vision and machine learning. Particularly he is interested in 3D perception for autonomous vehicles, including 3D detection and 3D segmentation.



Hongsheng Li is an assistant professor in the Department of Electronic Engineering at the Chinese University of Hong Kong. In 2013-2015, he was an associate professor in the School of Electronic Engineering at University of Electronic Science and Technology of China. He received the bachelor's degree in Automation from East China University of Science and Technology in 2006, and the doctorate degree in Computer Science from Lehigh University, United States in 2012. He has published over 70 papers in premier conferences on computer vision and machine learning, including CVPR, ICCV, ECCV, NeurIPS, ICLR, and AAAI. He received the 2020 IEEE CAS Society Outstanding Young Author Award. He won the first place in Object Detection from Videos (VID) track of ImageNet challenge 2016 as the team leader and 2015 as a team co-leader. He is an associate editor of Neurocomputing and serves as an area chair of NeurIPS 2021. His research interests include computer vision, machine learning, and medical image analysis.

Ziwei Liu is a Nanyang Assistant Professor at School of Computer Science and Engineering (SCSE) in Nanyang Technological University, with MMLab@NTU. Previously, he was a research fellow (2018-2020) in CUHK (with Prof. Dahua Lin) and a post-doc researcher (2017-2018) in UC Berkeley (with Prof. Stella X. Yu). His research interests include computer vision, machine learning and computer graphics. Ziwei received his Ph.D. (2013-2017) from CUHK / Multimedia Lab, advised by Prof. Xiaoou Tang and Prof. Xiaogang Wang. He is fortunate to have internships at Microsoft Research and Google Research. His works include Burst Denoising, CelebA, DeepFashion, Fashion Landmarks, DeepMRF, Voxel Flow, Long-Tailed Recognition, and Compound Domain Adaptation. His works have been transferred to products, including Microsoft Pix, SenseFocus, and Google Clips.