LidarMultiNet: Towards a Unified Multi-Task Network for LiDAR Perception

Dongqiangzi Ye[1*], Zixiang Zhou[1,2*†], Weijia Chen[1*], Yufei Xie[1*], Yu Wang[1], Panqu Wang[1], Hassan Foroosh[2]

[1] TuSimple
[2] University of Central Florida

Abstract

LiDAR-based 3D object detection, semantic segmentation, and panoptic segmentation are usually implemented in specialized networks with distinctive architectures that are difficult to adapt to each other. This paper presents **LidarMultiNet**, a LiDAR-based multi-task network that unifies these three major LiDAR perception tasks. Among its many benefits, a multi-task network can reduce the overall cost by sharing weights and computation among multiple tasks. However, it typically underperforms compared to independently combined single-task models. The proposed LidarMultiNet aims to bridge the performance gap between the multi-task network and multiple single-task networks. At the core of LidarMultiNet is a strong 3D voxel-based encoder-decoder architecture with a Global Context Pooling (GCP) module extracting global contextual features from a LiDAR frame. Task-specific heads are added on top of the network to perform the three LiDAR perception tasks. More tasks can be implemented simply by adding new task-specific heads while introducing little additional cost. A second stage is also proposed to refine the first-stage segmentation and generate accurate panoptic segmentation results. LidarMultiNet is extensively tested on both Waymo Open Dataset and nuScenes dataset, demonstrating for the first time that major LiDAR perception tasks can be unified in a single strong network that is trained end-to-end and achieves state-of-the-art performance. Notably, LidarMultiNet reaches the official **1st** place in the Waymo Open Dataset 3D semantic segmentation challenge 2022 with the highest mIoU and the best accuracy for most of the 22 classes on the test set, using only LiDAR points as input. It also sets the new state-of-the-art for a single model on the Waymo 3D object detection benchmark and three nuScenes benchmarks.

Introduction

LiDAR plays a major role in the field of autonomous driving. With the release of several large-scale multi-sensor datasets, (*e.g.* the Waymo Open Dataset (Sun et al. 2020) and the nuScenes (Caesar et al. 2020)) datasets, collected in real self-driving scenarios, LiDAR-based perception algorithms have significantly advanced in recent years. Thanks to the advancement of sparse convolution (Yan, Mao, and Li 2018; Choy, Gwak, and Savarese 2019), voxel-based LiDAR perception methods (Yin, Zhou, and Krähenbühl 2021) have become predominant on major 3D object detection and semantic segmentation benchmarks, and outperform their point-based, pillar-based, or projection-based counterparts (Fan et al. 2021; Lang et al. 2019; Qi et al. 2017) by a large margin in terms of both accuracy and efficiency. In voxel-based LiDAR perception networks, standard 3D sparse convolution is usually used in tandem with submanifold sparse convolution (Graham and van der Maaten 2017). Since standard 3D sparse convolution dilates the sparse features and increases the number of active sites, it is usually only applied as downsampling layer at each stage of the encoder followed by the submanifold sparse convolution layers. The submanifold sparse convolution maintains the number of active sites but it limits the information flow (Chen et al. 2022b) and the receptive field. However, a large receptive field is necessary to exploit the global contextual information, which is critical

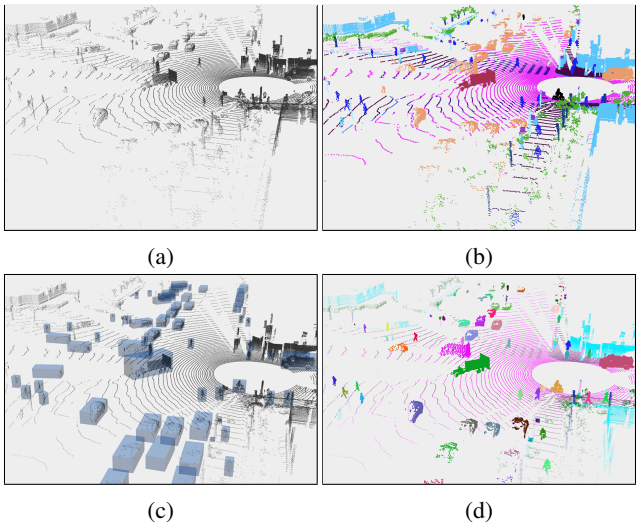

Figure 1: Our LidarMultiNet takes LiDAR point cloud (a) as input and performs simultaneous 3D semantic segmentation (b), 3D object detection (c), and panoptic segmentation (d) in a single unified network.

*These authors contributed equally.
†Work done during an internship at TuSimple.

for 3D segmentation tasks.

In LiDAR-based perception, 3D object detection, semantic segmentation, and panoptic segmentation are usually implemented in distinct and specialized network architectures (Yin, Zhou, and Krähenbühl 2021; Zhang et al. 2020; Zhou, Zhang, and Foroosh 2021; Zhu et al. 2021b; Cheng et al. 2021), which are task-specific and difficult to adapt to other LiDAR perception tasks. Multi-task networks (Teichmann et al. 2018; Feng et al. 2021), unify closely-related tasks by sharing the weights and computation among them, and therefore expected to improve the performance of individual tasks while reducing the overall computational cost. However, so far prior LiDAR multi-task networks have been underperforming compared to their single-task counterparts and have been failing to demonstrate state-of-the-art performance (Feng et al. 2021). As a result, single-task networks are still predominant in major LiDAR perception benchmarks. In this paper, we bridge the gap between the performance of single LiDAR multi-task networks and multiple independent task-specific networks. Specifically, we propose to unify 3D semantic segmentation, 3D object detection, and panoptic segmentation in a versatile network that exploits the synergy between these tasks and achieves state-of-the-art performance, as shown in Figure 1.

Our main contributions are four-fold, and are summarized below:

- We present a novel voxel-based LiDAR multi-task network that unifies three major LiDAR perception tasks and can be extended for new tasks with little increase in the computational cost by adding more task-specific heads.
- We propose a Global Context Pooling (GCP) module to improve the global feature learning in the encoder-decoder network based on 3D sparse convolution.
- We introduce a second-stage refinement module to refine the first-stage semantic segmentation of the foreground *thing* classes and produce accurate panoptic segmentation results.
- We demonstrate start-of-the-art performance for LidarMultiNet on 5 major LiDAR benchmarks. Notably, LidarMultiNet reaches the official 1$^{\text{st}}$ place in the Waymo 3D semantic segmentation challenge 2022. LidarMultiNet reaches the highest mAPH L2 for a single model on the Waymo 3D object detection benchmark. On the nuScenes semantic segmentation and panoptic segmentation benchmarks, LidarMultiNet outperforms the previously published state-of-the-art methods. On the nuScenes 3D object detection benchmark, LidarMultiNet sets a new standard for state-of-the-art performance in LiDAR-only non-ensemble methods.

Related Work

LiDAR Detection and Segmentation One key challenge for LiDAR perception is how to efficiently encode the large-scale sparsely distributed point cloud into a uniform feature representation. The common practice is transforming the point cloud into a discretized 3D or 2D map through a 3D voxelization (Zhou and Tuzel 2018; Zhu et al. 2021b), Bird's Eye View (BEV) projection (Yang, Luo, and Urtasun 2018; Lang et al. 2019; Zhang et al. 2020), or range-view projection (Wu et al. 2018; Sun et al. 2021). State-of-the-art LiDAR 3D object detectors (Yin, Zhou, and Krähenbühl 2021) typically project the 3D sparse tensor into a dense 2D BEV feature map and perform the detection on the BEV space. In contrast, LiDAR segmentation requires predicting the point-wise labels, hence a larger features map is needed to minimize the discretization error when projecting the voxel labels back to the points. Many methods (Tang et al. 2020; Xu et al. 2021; Ye et al. 2021b) also combine the point-level features with voxel features to retain the fine-grained features in a multi-view fusion manner.

In LiDAR-based 3D object detection, anchor-free detectors (Yin, Zhou, and Krähenbühl 2021) are predominant on major detection benchmarks and widely adopted for their efficiency. Our LidarMultiNet adopts the anchor-free 3D detection heads, which are attached to its 2D branch.

A second stage is often used in the detection framework (Shi et al. 2020; Yin, Zhou, and Krähenbühl 2021; Li, Wang, and Wang 2021; Sheng et al. 2021) to improve the detection accuracy through an RCNN-style network. It processes each object separately by extracting the features based on the initial bounding box prediction for refinement. LidarMultiNet adopts a second segmentation refinement stage based on the detection and segmentation results of the first stage.

LiDAR Panoptic Segmentation Recent LiDAR panoptic segmentation methods (Zhou, Zhang, and Foroosh 2021; Hong et al. 2021; Razani et al. 2021) usually derive from the well-studied segmentation networks (Zhang et al. 2020; Zhu et al. 2021b; Cheng et al. 2021) in a bottom-up fashion. This is largely due to the loss of height information in the detection networks, which makes them difficult to adjust the learned feature representation to the segmentation task. This results in two incompatible designs for the best segmentation (Xu et al. 2021) and detection (Yin, Zhou, and Krähenbühl 2021) methods. According to (Fong et al. 2022), end-to-end LiDAR panoptic segmentation methods still underperform compared to independently combined detection and segmentation models. In this work, our model can perform simultaneous 3D object detection and semantic segmentation and trains the tasks jointly in an end-to-end fashion.

Multi-Task Network Multi-task learning aims to unify multiple tasks into a single network and train them simultaneously in an end-to-end fashion. MultiNet (Teichmann et al. 2018) is a seminal work of image-based multi-task learning that unifies object detection and road understanding tasks in a single network. In LiDAR-based perception, LidarMTL (Feng et al. 2021) proposed a simple and efficient multi-task network based on 3D sparse convolution and deconvolutions for joint object detection and road understanding. In this work, we unify the major LiDAR-based perception tasks in a single, versatile, and strong network.

LidarMultiNet

Given a set of LiDAR point cloud $P = \{p_i | p_i \in \mathbb{R}^{3+c}\}_{i=1}^{N}$, where N is the number of points and each point has $3 + c$

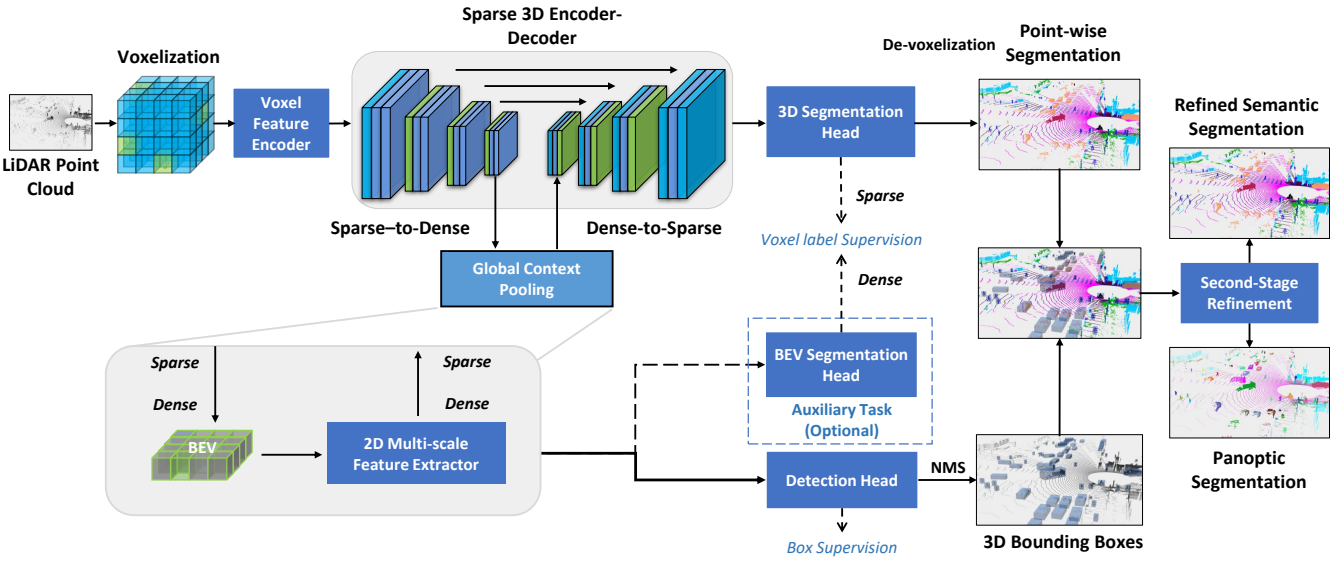

Figure 2: Main Architecture of the LidarMultiNet. At the core of our network is a 3D encoder-decoder based on 3D sparse convolution and deconvolutions. In between the encoder and the decoder, a Global Context Pooling (GCP) module is applied to extract contextual information through the conversion between sparse and dense feature maps and via a 2D multi-scale feature extractor. The 3D segmentation head is attached to the decoder and its predicted voxel labels are projected back to the point level via a de-voxelization step. Meanwhile, the 3D detection head and auxiliary BEV segmentation head are attached to the 2D BEV branch. The 2nd-stage produces the refined semantic segmentation and the panoptic segmentation results.

input features, the goals of the LiDAR object detection, semantic segmentation, and panoptic segmentation tasks are to predict the 3D bounding boxes, point-wise semantic labels L_{sem} of K classes, and panoptic labels L_{pan}, respectively. Compared to semantic segmentation, panoptic segmentation additionally requires the points in each instance to have a unique instance id.

Main Architecture

The main architecture of LidarMultiNet is illustrated in Figure 2. A voxelization step converts the original unordered LiDAR points to a regular voxel grid. A Voxel Feature Encoder (VFE) consisting of a Multi-Layer Perceptron (MLP) and max pooling layers is applied to generate enhanced sparse voxel features, which serve as the input to the 3D sparse U-Net architecture. Lateral skip-connected features from the encoder are concatenated with the corresponding voxel features in the decoder. A Global Context Pooling (GCP) (Ye et al. 2022) module with a 2D multi-scale feature extractor bridges the last encoder stage and the first decoder stage. 3D segmentation head is attached to the decoder and outputs voxel-level predictions, which can be projected back to the point level through the de-voxelization step. Heads of BEV tasks, such as 3D object detection, are attached to the 2D BEV branch. Given the detection and segmentation results of the first stage, the second stage is applied to refine semantic segmentation and generate panoptic segmentation results.

The 3D encoder consists of 4 stages of 3D sparse convolutions with increasing channel width. Each stage starts with a sparse convolution layer followed by two submanifold sparse convolution blocks. The first sparse convolution layer has a stride of 2 except at the first stage, therefore the spatial resolution is downsampled by 8 times in the encoder. The 3D decoder also has 4 symmetrical stages of 3D sparse deconvolution blocks but with decreasing channel width except for the last stage. We use the same sparse convolution key indices between the encoder and decoder layers to keep the same sparsity of the 3D voxel feature map.

For the 3D object detection task, we adopt the detection head of the anchor-free 3D detector CenterPoint (Yin, Zhou, and Krähenbühl 2021) and attach it to the 2D multi-scale feature extractor. Besides the detection head, an additional BEV segmentation head also can be attached to the 2D branch of the network, providing coarse segmentation results and serving as an auxiliary loss during the training.

Global Context Pooling

3D sparse convolution drastically reduces the memory consumption of the 3D CNN for the LiDAR point cloud data, but it generally requires the layers of the same scale to retain the same sparsity in both encoder and decoder. This restricts the network to use only submanifold convolution (Graham, Engelcke, and Van Der Maaten 2018) in the same scale. However, submanifold convolution cannot broadcast features to isolated voxels through stacking multiple convolution layers. This limits the ability of CNN to learn long-range global information. Inspired by the Region Proposal Network (RPN) (Ren et al. 2015) in the 3D detection network, we design a Global Context Pooling (GCP) module

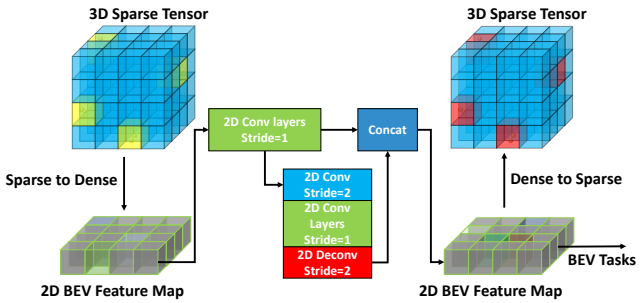

Figure 3: Illustration of the Global Context Pooling (GCP) module. 3D sparse tensor is projected to a 2D BEV feature map. Two levels of 2D BEV feature maps are concatenated and then converted back to a 3D sparse tensor, which serves as the input to the BEV task heads.

to extract large-scale information through a dense BEV feature map. On the one hand, GCP can efficiently enlarge the receptive field of the network to learn global contextual information for the segmentation task. On the other hand, its 2D BEV dense feature can also be used for 3D object detection or other BEV tasks, by attaching task-specific heads with marginal additional computational cost.

As illustrated in Figure 3, given the low-resolution feature representation of the encoder output, we first transform the sparse voxel feature into a dense feature map $\mathcal{F}^{sparse}_{encoder} \in \mathbb{R}^{C' \times M'} \rightarrow \mathcal{F}^{dense} \in \mathbb{R}^{C' \times \frac{D}{d_z} \times \frac{H}{d_x} \times \frac{W}{d_y}}$, where d is the downsampling ratio and M' is the number of valid voxels in the last scale. We concatenate the features in different heights together to form a 2D BEV feature map $\mathcal{F}^{bev}_{in} \in \mathbb{R}^{(C' * \frac{D}{d_z}) \times \frac{H}{d_x} \times \frac{W}{d_y}}$. Then, we use a 2D multi-scale CNN to further extract long-range contextual information. Note that we can utilize a deeper and more complex structure with a trivial run-time overhead, since the BEV feature map has a relatively small resolution. Lastly, we reshape the encoded BEV feature representation to the dense voxel map, then transform it to the sparse voxel feature following the reverse dense to sparse conversion.

Benefiting from GCP, our architecture could significantly enlarge the receptive field, which plays an important role in semantic segmentation. In addition, the BEV feature maps in GCP can be shared with other tasks (eg. object detection) simply by attaching additional heads with slight increase of computational cost. By utilizing the BEV-level training like object detection, GCP can enhance the segmentation performance furthermore.

Multi-task Training and Losses

The 3D segmentation branch predicts voxel-level labels $L^v = \{l_j | l_j \in (1 \dots K)\}_{j=1}^{M}$ given the learned voxel features $\mathcal{F}^{sparse}_{decoder} \in \mathbb{R}^{C \times M}$ output by the 3D decoder. M stands for the number of active voxels in the output and C represents the dimension of every output feature. We supervise it through a combination of cross-entropy loss and Lovasz loss (Berman, Rannen Triki, and Blaschko 2018): $\mathcal{L}_{SEG} = \mathcal{L}^v_{ce} + \mathcal{L}^v_{Lovasz}$. Note that $\mathcal{L}_{SEG}$ is a sparse loss, and the computational cost as well as the GPU memory usage are much smaller than dense loss.

The detection heads are applied on the 2D BEV feature map: $\mathcal{F}^{bev}_{out} \in \mathbb{R}^{C_{bev} \times \frac{H}{d_x} \times \frac{W}{d_y}}$. They predict a class-specific heatmap, the object dimensions and orientation, and a IoU rectification score, which are supervised by the focal loss (Lin et al. 2017) ($\mathcal{L}_{hm}$) and L1 loss ($\mathcal{L}_{reg}, \mathcal{L}_{iou}$) respectively: $\mathcal{L}_{DET} = \lambda_{hm}\mathcal{L}_{hm} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{iou}\mathcal{L}_{iou}$, where the weights $\lambda_{hm}, \lambda_{reg}, \lambda_{iou}$ are empirically set to $[1, 2, 1]$.

During training, the BEV segmentation head is supervised with $\mathcal{L}_{BEV}$, a dense loss consisting of cross-entropy loss and Lovasz loss: $\mathcal{L}_{BEV} = \mathcal{L}^{bev}_{ce} + \mathcal{L}^{bev}_{Lovasz}$.

Our network is trained end-to-end for multiple tasks. Similar to (Feng et al. 2021), we define the weight of each component of the final loss based on the uncertainty (Kendall, Gal, and Cipolla 2018) as follows:

$$\mathcal{L}_{total} = \sum_{L_i \in \left\{ \substack{L^v_{ce}, L^v_{Lovasz}, L^{bev}_{ce}, \\ L^{bev}_{Lovasz}, L_{hm}, L_{reg}, L_{iou}} \right\}} \frac{1}{2\sigma_i^2}\mathcal{L}_i + \frac{1}{2}\log\sigma_i^2 \tag{1}$$

where σ_i is the learned parameter representing the degree of uncertainty in $task_i$. The more uncertain the $task_i$ is, the less $\mathcal{L}_i$ contributes to $\mathcal{L}_{total}$. The second part can be treated as a regularization term for σ_i during training.

Instead of assigning an uncertainty-based weight to every single loss, we first group the losses belonging to the same task with fixed weights. The resulting three task-specific losses (*i.e.*, $\mathcal{L}_{SEG}$, $\mathcal{L}_{DET}$, $\mathcal{L}_{BEV}$) are then combined using weights defined based on the uncertainty:

$$\mathcal{L}_{total} = \sum_{i \in \{SEG, DET, BEV\}} \frac{1}{2\sigma_i^2}\mathcal{L}_i + \frac{1}{2}\log\sigma_i^2 \tag{2}$$

Second-stage Refinement

Coarse panoptic segmentation result can be obtained directly by fusing the first-stage semantic segmentation and object detection results, *i.e.*, assigning a unique ID to the points classified as one of the foreground *thing* classes within a 3D bounding box. However, the points within a detected bounding box can be misclassified as multiple classes due to the lack of spatial prior knowledge, as shown in Figure 5. In order to improve the spatial consistency for the *thing* classes, we propose a novel point-based approach as the second stage to refine the first-stage segmentation and provide accurate panoptic segmentation.

The second stage is illustrated in Figure 4. Specifically, it takes features from raw point cloud P, the B predicted bounding boxes, sparse voxel features $\mathcal{F}^{sparse}_{decoder}$, and the BEV feature map $\mathcal{F}^{bev}_{out}$ to predict box classification scores S_{box} and point-wise mask scores S_{point}. Given the B bounding box predictions in the 1st stage, we first transform each point within a box into its local coordinates. Then we concatenate its local coordinates with the corresponding voxel features from $\mathcal{F}^{sparse}_{decoder}$. Meanwhile, we extract 2nd-stage box-wise features as in (Yin, Zhou, and Krähenbühl 2021) from $\mathcal{F}^{bev}$. We assign a point-box index $I = \{ind_i | ind_i \in$

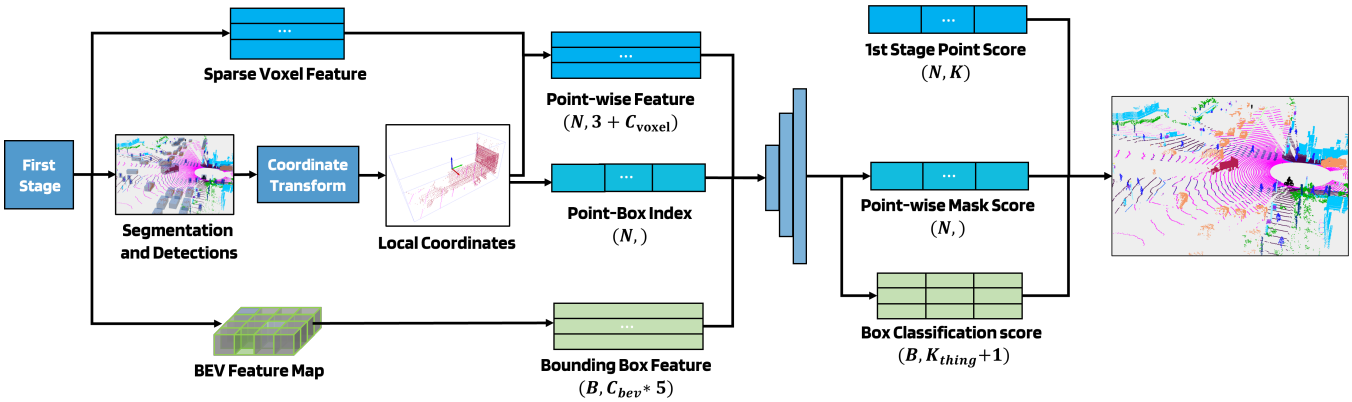

Figure 4: Illustration of the second-stage refinement pipeline. The architecture of the second-stage refinement is point-based. We first fuse the detected boxes, voxel-wise features, and BEV features from the 1st stage to generate the inputs for the 2nd stage. The local coordinate transformation is applied to the points within each box. Then, a point-based backbone with MLPs, attention modules, and aggregation modules infer the box classification scores and point-wise mask scores. The final refined segmentation scores are computed by fusing the 1st and 2nd stage predictions.

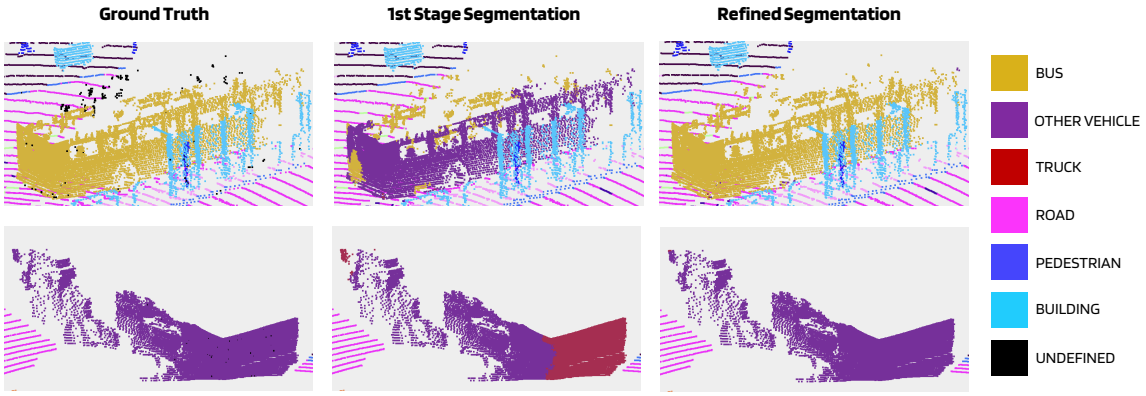

Figure 5: Examples of the 2nd-stage refinement. The segmentation consistency of points of the thing objects can be improved by the 2nd stage.

$\mathbb{I}, 0 \leq ind_i \leq B\}_{i=1}^{N}$ to the points in each box. The points that are not in any boxes are assigned with index ϕ and will not be refined in the 2nd stage. Next, we use a PointNet-like network to predict point-wise mask scores $S_{point} = \{sp_i | sp_i \in (0,1)\}_{i=1}^{N}$ and box classification scores $S_{box} = \{sb_i | sb_i \in (0,1)^{K_{thing}+1}\}_{i=1}^{B}$, where K_{thing} denotes the number of *thing* classes and the one additional class represents the remaining *stuff* classes $\emptyset$. During training, we supervise the box-wise class scores through a cross-entropy loss and the point-wise mask scores through a binary cross-entropy loss.

We merge the 2nd-stage predictions with the 1st-stage semantic scores to generate the final semantic segmentation predictions $\hat{L}_{sem}$. To refine segmentation score $S_{2nd} = \{rs_i | rs_i \in (0,1)^{K_{thing}+1}\}_{i=1}^{N}$, we combine the point-wise mask scores with their corresponding box-wise class scores as follows:

$$S_{2nd}(j) = \begin{cases} S_{point} \times S_{box}(j), & j \in \mathcal{I}^{K_{thing}} \\ S_{point} \times S_{box}(j) + S'_{point}, & j = \emptyset \end{cases} \tag{3}$$

where K_{thing} denotes the number of *thing* classes, $\emptyset$ denotes the rest *stuff* classes which would not be refined in the 2nd stage, $S_{point} = \{sp_i | sp_i \in (0,1)\}_{i=1}^{N}$ is the point-wise mask scores, $S'_{point} = 1 - S_{point}$ and $S_{box} = \{sb_i | sb_i \in (0,1)^{K_{thing}+1}\}_{i=1}^{B}$ is the box classification scores. N and B denote the number of points and boxes.

In addition, the points not in any boxes can be considered as $S_{2nd}(\emptyset) = 1$, which means their scores are the same as the 1st-stage scores. We then further combine the refined scores with the 1st-stage scores as follows:

$$S_{final} = \begin{cases} S_{1st} \times S_{2nd}(\emptyset) + S_{2nd}(*), & ind_i \neq \phi, * \neq \emptyset \\ S_{1st}, & ind_i = \phi \end{cases} \tag{4}$$

where ϕ denotes the index where points are not in any boxes, and $S_{1st} = \{sf_i | sf_i \in (0,1)^K\}_{i=1}^N$ is the 1st stage scores.

The scores S_{final} are used to generate the semantic segmentation results $\hat{L}_{sem}$ through finding the class with the maximum score. It is intuitive to infer the final panoptic segmentation results through the 1st-stage boxes and the final semantic segmentation results S_{box} and $\hat{L}_{sem}$. First, we extract points for a box where points and the box have the same semantic category. Then the extracted points will be assigned a unique index as the instance id for the panoptic segmentation.

Experiments

In this section, we perform extensive tests of the proposed LidarMultiNet on five major benchmarks of the large-scale Waymo Open Dataset (Sun et al. 2020) (*3D Object Detection* and *3D Semantic Segmentation*) and nuScenes dataset (Caesar et al. 2020; Fong et al. 2022) (*Detection*, *LiDAR Segmentation*, and *Panoptic Segmentation*). More details are included in the arXiv version[1].

Datasets and Metrics

Waymo Open Dataset (WOD) contains 1150 sequences in total, split into 798 in the training set, 202 in the validation set, and 150 in the test set. Each sequence contains about 200 frames of LiDAR point cloud captured at 10 FPS with multiple LiDAR sensors. Object bounding box annotations are provided in each frame while the 3D semantic segmentation labels are provided only for sampled frames. WOD uses Average Precision Weighted by Heading (APH) as the main evaluation metric for the detection task. There are two levels of difficulty, LEVEL_2 (L2) is assigned to examples where either the annotators label as hard or if the example has less than 5 LiDAR points, while LEVEL_1 (L1) is assigned to the rest of the examples. Both L1 and L2 examples participate in the computation of the primary metric mAPH L2.

For the semantic segmentation task, we use the v1.3.2 dataset, which contains 23,691 and 5,976 frames with semantic segmentation labels in the training set and validation set, respectively. There are a total of 2,982 frames in the final test set. WOD has semantic labels for a total of 23 classes, including an undefined class. Intersection Over Union (IOU) metric is used as the evaluation metric.

NuScenes contains 1000 scenes with 20 seconds duration each, split into 700 in the training set, 150 in the validation set, and 150 in the test set. The sensor suite contains a 32-beam LiDAR with 20Hz capture frequency. For the object detection task, the key samples are annotated at 2Hz with ground truth labels for 10 foreground object classes (*thing*). For the semantic segmentation and panoptic segmentation tasks, every point in the keyframe is annotated using 6 more background classes (*stuff*) in addition to the 10 *thing* classes. NuScenes uses mean Average Precision (mAP) and NuScenes Detection Score (NDS) metrics for the detection task, mIoU and Panoptic Quality (PQ) (Kirillov et al. 2019) metrics for the semantic and panoptic segmentation.

Implementation Details

On the Waymo Open Dataset, the point cloud range is set to $[-75.2m, 75.2m]$ for x axis and y axis, and $[-2m, 4m]$ for the z axis, and the voxel size is set to $(0.1m, 0.1m, 0.15m)$. Following (Yin, Zhou, and Krähenbühl 2021), we transform the past two LiDAR frames using the vehicle's pose information and merge them with the current LiDAR frame to produce a denser point cloud and append a timestamp feature to each LiDAR point. Points of past LiDAR frames participate in the voxel feature computation but do not contribute to the loss calculation.

On the nuScenes dataset, the point cloud range is set to $[-54m, 54m]$ for x axis and y axis, and $[-5m, 3m]$ for the z axis, and the voxel size is set to $(0.075m, 0.075m, 0.2m)$. Following the common practice (Caesar et al. 2020; Yin, Zhou, and Krähenbühl 2021) on nuScenes, we transform and concatenate points from the past 9 frames with the current point cloud to generate a denser point cloud. Following (Yin, Zhou, and Krähenbühl 2021), we apply separate detection heads in the detection branch for different categories.

During training, we employ data augmentation which includes standard random flipping, and global scaling, rotation and translation. We also adopt the ground-truth sampling (Yan, Mao, and Li 2018) with the fade strategy (Wang et al. 2021). We train the models using AdamW (Loshchilov and Hutter 2017) optimizer with one-cycle learning rate policy, with a max learning rate of 3e-3, a weight decay of 0.01, and a momentum ranging from 0.85 to 0.95. We use a batch size of 2 on each of the 8 A100 GPUs. For the one-stage model, we train the models from scratch for 20 epochs. For the two-stage model, we freeze the 1st stage and finetune the 2nd stage for 6 epochs.

Waymo Open Dataset Results

3D Semantic Segmentation Challenge Leaderboard We tested the performance of LidarMultiNet on the WOD 3D Semantic Segmentation Challenge. Since the semantic segmentation challenge only considers semantic segmentation accuracy, our model is trained with focus on semantic segmentation, while object detection and BEV segmentation both serve as the auxiliary tasks. Since there is no runtime constraint, most participants employed the Test-Time Augmentation (TTA) and model ensemble to further improve the performance of their methods. For details regarding the TTA and ensemble, please refer to the supplementary material. Table 1 is the final WOD semantic segmentation leaderboard and shows that our LidarMultiNet (Ye et al. 2022) achieves a mIoU of 71.13 and ranks the 1st place on the leaderboard[1], and also has the best IoU for 15 out of the total 22 classes. Note that our LidarMultiNet uses only LiDAR point cloud as input, while some other entries on the leaderboard (e.g. SegNet3DV2) use both LiDAR points and camera images

[1]https://arxiv.org/abs/2209.09385

[1]https://waymo.com/open/challenges/2022/3d-semantic-segmentation, accessed on August 06, 2022.

Waymo Leaderboard	mIoU	*CAR*	*TRUCK*	*BUS*	*PED*	*SIGN*	*BIKE*	*MTRCL*	*BLDG*	*VEG*	*ROAD*	*SW*
LidarMultiNet	**71.13**	**95.86**	**70.57**	81.44	**93.23**	**73.84**	69.92	76.73	**97.37**	**88.92**	93.21	**87.95**
LidarMultiNet†	69.69	95.47	65.36	80.44	92.70	71.82	69.30	76.78	97.17	88.47	92.72	86.75
SegNet3DV2	70.48	95.73	69.03	79.74	92.66	71.82	69.53	**76.97**	97.15	88.18	**93.27**	87.25
HorizonSegExpert	69.44	95.55	68.93	**84.38**	92.22	72.71	68.56	71.7	97.18	87.81	93.09	87.29
HRI_HZ_SMRPV	69.38	95.79	66.97	78.36	91.83	72.93	**70.23**	76.02	96.94	87.2	92.6	86.03
Waymo_3DSEG	68.99	95.47	69.54	77.85	91.12	72.73	68.87	74.87	96.96	87.89	92.61	86.13
SPVCNN++	67.7	95.12	67.73	75.61	91.57	73.24	65.97	73.6	90.82	86.58	91.73	85.87
PolarFuse	67.28	95.06	67.79	77.04	89.61	65.37	62.97	67.96	96.38	87.07	92.54	85.02
LeapNet	66.89	94.45	65.68	79.01	90.07	70.46	61.58	66.84	96.65	87.01	92.61	86.34
3DSEG	66.77	94.64	66.95	77.61	89.18	70.01	60.81	67.91	96.55	87.33	91.79	85.04
CAVPercep	63.73	93.64	62.83	68.12	87.39	67.03	55.6	60.84	96.15	86.48	91.24	83.15
VS-Concord3D	63.54	92.6	66.9	73.13	85.51	66.88	62.26	68.11	86.04	75.13	90.6	82.96
RGBV-RP Net	62.61	94.87	67.48	74.91	88.66	68.05	53.82	64.92	96.59	86.47	90.97	84.48

Table 1: Waymo Open Dataset Semantic Segmentation Leaderboard. Results of more classes can be found in the arXiv version. †: without TTA and model ensemble.

Baseline	Multi-frame	GCP	$\mathcal{L}_{BEV}$	$\mathcal{L}_{DET}$	Two-Stage	TTA	Ensemble	mIoU
✓								69.90
✓	✓							70.49
✓	✓	✓						71.43
✓	✓	✓	✓					71.58
✓	✓	✓	✓	✓				72.06
✓	✓	✓	✓	✓	✓			72.40
✓	✓	✓	✓	✓	✓	✓		73.05
✓	✓	✓	✓	✓	✓	✓	✓	73.78

Table 2: Ablation studies for 3D semantic segmentation on the WOD *validation* set.

and therefore require running additional 2D CNNs to extract image features. For a better reference, we also test the result of LidarMultiNet that is trained with both detection and segmentation as the main tasks. LidarMultiNet reaches a mIoU of 69.69 on WOD 3D segmentation test set without TTA and model ensemble.

Ablation Study on the 3D Semantic Segmentation Validation Set We ablate each component of the LidarMultiNet and the results on the 3D semantic segmentation validation set are shown in Table 2. Our baseline network reaches a mIoU of 69.90 on the validation set. On top of this baseline, multi-frame input (*i.e.*, including the past two frames) brings a 0.59 mIoU improvement. The GCP further improves the mIoU by 0.94. The auxiliary losses, (*i.e.*, BEV segmentation and 3D object detection) result in a total improvement of 0.63 mIoU, and the 2nd-stage improves the mIoU by 0.34, forming our best single model on the WOD validation set. TTA and ensemble further improve the mIoU to 73.05 and 73.78, respectively.

Evaluation on the 3D Object Detection Benchmark To demonstrate that LidarMultiNet can outperform single-task models on both detection and segmentation tasks, we tested it on the WOD 3D object detection benchmark and compared with state-of-the-art 3D object detection methods. The model is trained with both detection and segmentation as the main tasks, and its detection head is trained for detecting three classes, (*i.e.* vehicle, pedestrian, and cyclist). The model is trained for 20 epochs with the fade strategy, *i.e.*, ground-truth sampling for the object detection task is disabled for the last 5 epochs. The results on WOD test set are shown in Table 3. Our LidarMultiNet method reaches the highest mAPH L2 of 76.35 on the test set for a single model without TTA and outperforms the state-of-art 3D object detectors, including the multi-modal detectors which also leverage camera information. LidarMultiNet also outperforms other multi-frame fusion methods that require more past frames. Moreover, the same LidarMultiNet model reaches a mIoU of 71.93 on the WOD semantic segmenta-

Methods	Modal	Frames	mAPH L2
PV-RCNN++ (2022)	L	1	70.20
CenterPoint++ (2021)	L	3	72.82
SST_3f (2022)	L	3	72.81
AFDetV2 (2022)	L	2	73.12
DeepFusion (2022b)	CL	5	75.54
MPPNet (2022a)	L	16	75.67
CenterFormer (2022)	L	16	76.29
BEVFusion (2023)	CL	3	76.33
LidarMultiNet (Ours)	L	3	**76.35**

Table 3: Single-model detection performance comparisons on Waymo *test* set. "L" indicates LiDAR-only, and "CL" denotes camera and LiDAR fusion.

Models	L2 mAPH	mIoU
Detection-only	75.01	-
Segmentation-only	-	71.58
Joint Training	75.15	71.93

Table 4: Comparison of the first-stage result of the jointly trained model and indepedently trained single-task models.

	Methods	mIoU	NDS	PQ
Segmentation	PolarNet (2020)	69.8	-	-
	PolarStream (2021)	73.4	-	-
	JS3C-Net (2021)	73.4	-	-
	AMVNet (2021)	77.3	-	-
	SPVNAS (2020)	77.4	-	-
	Cylinder3D++ (2021b)	77.9	-	-
	AF2S3Net (2021)	78.3	-	-
	DRINet++ (2021a)	80.4	-	-
	SPVCNN++ (2020)	81.1	-	-
Detection	Cylinder3D (2021a)	-	61.6	-
	CBGS (2019)	-	63.3	-
	CenterPoint (2021)	-	65.5	-
	HotSpotNet (2020)	-	66.0	-
	Object DGCNN (2021)	-	66.1	-
	AFDetV2 (2022)	-	68.5	-
	Focals Conv (2022b)	-	70.0	-
	TransFusion-L (2022)	-	70.2	-
	LargeKernel3D (2022c)	-	70.5	-
Panoptic	EfficientLPS (2021)	-	-	62.4
	Panoptic-PolarNet (2021)	-	-	63.6
	SPVCNN++ (2020)	-	-	79.1
	Panoptic-PHNet (2022a)	-	-	80.1
	LidarMultiNet	**81.4**	**71.6**	**81.4**

Table 5: Comparison with state-of-the-art methods on the *test* sets of three nuScenes benchmarks. A single LidarMultiNet model is used to generate predictions for all three tasks.

tion validation set. In comparison, the other detectors and segmentation methods on the WOD benchmarks are all single-task models dedicated for either object detection or semantic segmentation.

Effect of the Joint Multi-task Training Table 4 shows an ablation study comparing the first-stage result of the jointly-trained model with independently-trained models. The segmentation-only model removes the detection head, while keeping only the 3D segmentation head and the BEV segmentation head. The detection-only model keeps only the detection head. Compared to the single-task models, the jointly-trained model performs better on both segmentation and detection. In addition, by sharing part of the network among the tasks, the jointly-trained model is also more efficient than directly combining the independent single-task models.

NuScenes Benchmarks

Comparison with State-of-the-art Methods On the nuScenes detection, semantic segmentation, and panoptic segmentation benchmarks, we compare LidarMultiNet with state-of-the-art LiDAR-based methods. The test set results of all three benchmarks are summarized in Table 5, respectively. As shown in the tables, a single-model LidarMultiNet without TTA outperforms the previous state-of-the-art methods on each task. Combining the independently trained state-of-the-art single-task models (*i.e.*, DRINet++, LargeKernel3D, and Panoptic-PHNet) reaches 80.4 mIOU, 70.5 NDS, and 80.2 PQ on the test set. In comparison, one single LidarMultiNet model without TTA outperforms their combined performance by 1.0% in mIOU, 1.1% in NDS, and 1.3% in PQ.

In summary, to the best of our knowledge, LidarMultiNet is the first time that a single LiDAR multi-task model surpasses previous single-task state-of-the-art methods for the three major LiDAR-based perception tasks.

Stage	Semantic	Panoptic			
	mIoU	PQ	SQ	RQ	mIoU
1st stage	81.7	81.2	90.5	89.3	83.3
2nd stage	82.0	81.8	90.8	89.7	83.6

Table 6: Improvements of the 2nd-stage segmentation refinement on the nuScenes semantic segmentation and panoptic segmentation *validation* sets.

Effect of the Second-Stage Refinement An ablation study on the effect of the proposed 2nd stage is shown in Table 6. With the 1st-stage detection and semantic predictions, LidarMultiNet already can get high panoptic segmentation results by directly fusing these two results together. The proposed 2nd stage further improves both semantic segmentation and panoptic segmentation results.

Top-down LiDAR Panoptic Segmentation CNN-based top-down panoptic segmentation methods have shown competitive performance compared to the bottom-up methods in the image domain. However, most previous LiDAR panoptic segmentation methods (Zhou, Zhang, and Foroosh 2021; Li et al. 2022a) adopt the bottom-up design due to the need of an accurate semantic prediction. And the cumbersome network structures with multi-view or point-level features fusion make them difficult to perform well in the object detection task. On the other hand, thanks to the GCP module and joint training design, LidarMultiNet can reach top performance on both object detection and semantic segmentation tasks. Even without a dedicated panoptic head, LidarMultiNet already outperforms the previous state-of-the-art bottom-up method.

Conclusion

We present the LidarMultiNet , which reached the official 1st place in the Waymo Open Dataset 3D semantic segmentation challenge 2022. LidarMultiNet is the first multi-task network to achieve state-of-the-art performance on all five major large-scale LiDAR perception benchmarks. We hope our LidarMultiNet can inspire future works in the unification of all LiDAR perception tasks in a single, versatile, and strong multi-task network.

References

Bai, X.; Hu, Z.; Zhu, X.; Huang, Q.; Chen, Y.; Fu, H.; and Tai, C.-L. 2022. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*.

Berman, M.; Rannen Triki, A.; and Blaschko, M. B. 2018. The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. Nuscenes: A multimodal dataset for autonomous driving. In *CVPR*.

Chen, Q.; Sun, L.; Wang, Z.; Jia, K.; and Yuille, A. 2020. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In *ECCV*.

Chen, Q.; Vora, S.; and Beijbom, O. 2021. Polarstream: Streaming object detection and segmentation with polar pillars. In *NeurIPS*.

Chen, X.; Shi, S.; Zhu, B.; Cheung, K. C.; Xu, H.; and Li, H. 2022a. MPPNet: Multi-Frame Feature Intertwining with Proxy Points for 3D Temporal Object Detection. In *ECCV*.

Chen, Y.; Li, Y.; Zhang, X.; Sun, J.; and Jia, J. 2022b. Focal Sparse Convolutional Networks for 3D Object Detection. In *CVPR*.

Chen, Y.; Liu, J.; Qi, X.; Zhang, X.; Sun, J.; and Jia, J. 2022c. Scaling up Kernels in 3D CNNs. In *arXiv:2206.10555*.

Cheng, R.; Razani, R.; Taghavi, E.; Li, E.; and Liu, B. 2021. (AF)2-S3Net: Attentive Feature Fusion With Adaptive Feature Selection for Sparse Semantic Segmentation Network. In *CVPR*.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *ICCV*.

Fan, L.; Pang, Z.; Zhang, T.; Wang, Y.-X.; Zhao, H.; Wang, F.; Wang, N.; and Zhang, Z. 2022. Embracing single stride 3d object detector with sparse transformer. In *CVPR*.

Fan, L.; Xiong, X.; Wang, F.; Wang, N.; and Zhang, Z. 2021. Rangedet: In defense of range view for lidar-based 3d object detection. In *ICCV*.

Feng, D.; Zhou, Y.; Xu, C.; Tomizuka, M.; and Zhan, W. 2021. A Simple and Efficient Multi-task Network for 3D Object Detection and Road Understanding. In *IROS*.

Fong, W. K.; Mohan, R.; Hurtado, J. V.; Zhou, L.; Caesar, H.; Beijbom, O.; and Valada, A. 2022. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. In *RA-L*.

Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*.

Graham, B.; and van der Maaten, L. 2017. Submanifold Sparse Convolutional Networks. In *arXiv:1706.01307*.

Hong, F.; Zhou, H.; Zhu, X.; Li, H.; and Liu, Z. 2021. Lidar-based panoptic segmentation via dynamic shifting network. In *CVPR*.

Hu, Y.; Ding, Z.; Ge, R.; Shao, W.; Huang, L.; Li, K.; and Liu, Q. 2022. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *AAAI*.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019. Panoptic segmentation. In *CVPR*.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. PointPillars: Fast encoders for object detection from point clouds. In *CVPR*.

Li, J.; He, X.; Wen, Y.; Gao, Y.; Cheng, X.; and Zhang, D. 2022a. Panoptic-PHNet: Towards Real-Time and High-Precision LiDAR Panoptic Segmentation via Clustering Pseudo Heatmap. In *CVPR*.

Li, Y.; Yu, A. W.; Meng, T.; Caine, B.; Ngiam, J.; Peng, D.; Shen, J.; Lu, Y.; Zhou, D.; Le, Q. V.; et al. 2022b. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *CVPR*.

Li, Z.; Wang, F.; and Wang, N. 2021. Lidar r-cnn: An efficient and universal 3d object detector. In *CVPR*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*.

Liong, V. E.; Nguyen, T. N. T.; Widjaja, S.; Sharma, D.; and Chong, Z. J. 2021. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. In *IJCAIW*.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.; and Han, S. 2023. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *ICRA*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. In *ICLR*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.

Razani, R.; Cheng, R.; Li, E.; Taghavi, E.; Ren, Y.; and Bingbing, L. 2021. Gp-s3net: Graph-based panoptic sparse semantic segmentation network. In *ICCV*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Sheng, H.; Cai, S.; Liu, Y.; Deng, B.; Huang, J.; Hua, X.-S.; and Zhao, M.-J. 2021. Improving 3D Object Detection With Channel-Wise Transformer. In *ICCV*.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*.

Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; and Li, H. 2022. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection. In *IJCV*.

Sirohi, K.; Mohan, R.; Büscher, D.; Burgard, W.; and Valada, A. 2021. Efficientlps: Efficient lidar panoptic segmentation. In *T-RO*.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*.

Sun, P.; Wang, W.; Chai, Y.; Elsayed, G.; Bewley, A.; Zhang, X.; Sminchisescu, C.; and Anguelov, D. 2021. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *CVPR*.

Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; and Han, S. 2020. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*.

Teichmann, M.; Weber, M.; Zoellner, M.; Cipolla, R.; and Urtasun, R. 2018. MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. In *IV*.

Wang, C.; Ma, C.; Zhu, M.; and Yang, X. 2021. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*.

Wang, Y.; and Solomon, J. M. 2021. Object dgcnn: 3d object detection using dynamic graphs. In *NeurIPS*.

Wu, B.; Wan, A.; Yue, X.; and Keutzer, K. 2018. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*.

Xu, J.; Zhang, R.; Dou, J.; Zhu, Y.; Sun, J.; and Pu, S. 2021. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *ICCV*.

Yan, X.; Gao, J.; Li, J.; Zhang, R.; Li, Z.; Huang, R.; and Cui, S. 2021. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. In *Sensors*.

Yang, B.; Luo, W.; and Urtasun, R. 2018. Pixor: Real-time 3d object detection from point clouds. In *CVPR*.

Ye, D.; Chen, W.; Zhou, Z.; Xie, Y.; Wang, Y.; Wang, P.; and Foroosh, H. 2022. LidarMultiNet: Unifying LiDAR Semantic Segmentation, 3D Object Detection, and Panoptic Segmentation in a Single Multi-task Network. In *arXiv:2206.11428*.

Ye, M.; Wan, R.; Xu, S.; Cao, T.; and Chen, Q. 2021a. DRINet++: Efficient Voxel-as-point Point Cloud Segmentation. In *arXiv:2111.08318*.

Ye, M.; Xu, S.; Cao, T.; and Chen, Q. 2021b. Drinet: A dual-representation iterative learning network for point cloud segmentation. In *ICCV*.

Yin, T.; Zhou, X.; and Krähenbühl, P. 2021. Center-based 3D Object Detection and Tracking. In *CVPR*.

Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; and Foroosh, H. 2020. PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation. In *CVPR*.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*.

Zhou, Z.; Zhang, Y.; and Foroosh, H. 2021. Panoptic-PolarNet: Proposal-Free LiDAR Point Cloud Panoptic Segmentation. In *CVPR*.

Zhou, Z.; Zhao, X.; Wang, Y.; Wang, P.; and Foroosh, H. 2022. CenterFormer: Center-based Transformer for 3D Object Detection. In *ECCV*.

Zhu, B.; Jiang, Z.; Zhou, X.; Li, Z.; and Yu, G. 2019. Class-balanced grouping and sampling for point cloud 3d object detection. In *arXiv:1908.09492*.

Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Li, W.; Ma, Y.; Li, H.; Yang, R.; and Lin, D. 2021a. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR-based Perception. In *TPMI*.

Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Ma, Y.; Li, W.; Li, H.; and Lin, D. 2021b. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*.