# LidarMultiNet: Unifying LiDAR Semantic Segmentation, 3D Object Detection, and Panoptic Segmentation in a Single Multi-task Network

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

{eowinye, weijia.chen619, zhouzixiang130, univerflyxie, yuwangrpi, wangpanqumanu}@gmail.com
Hassan.Foroosh@ucf.edu

## Abstract

*This technical report presents the 1st place winning solution for the Waymo Open Dataset 3D semantic segmentation challenge 2022. Our network, termed **LidarMulti-Net**, unifies the major LiDAR perception tasks such as 3D semantic segmentation, object detection, and panoptic segmentation in a single framework. At the core of LidarMulti-Net is a strong 3D voxel-based encoder-decoder network with a novel Global Context Pooling (GCP) module extracting global contextual features from a LiDAR frame to complement its local features. An optional second stage is proposed to refine the first-stage segmentation or generate accurate panoptic segmentation results. Our solution achieves a mIoU of 71.13 and is the best for most of the 22 classes on the Waymo 3D semantic segmentation test set, outperforming all the other 3D semantic segmentation methods on the official leaderboard [7]. We demonstrate for the first time that major LiDAR perception tasks can be unified in a single strong network that can be trained end-to-end.*

## 1. Introduction

LiDAR 3D semantic segmentation is a fundamental perception task of autonomous driving. With the recent release of several large-scale LiDAR point cloud datasets with semantic labels [1, 3, 25], more methods are proposed to advance the research in LiDAR semantic segmentation.

Compared to 2D image and 3D indoor point cloud segmentation, outdoor LiDAR point cloud presents more challenges for the segmentation problem. Due to the large-scale and sparsity of LiDAR point clouds, well-studied 2D and indoor 3D semantic segmentation methods [19, 20] fail to directly adjust to LiDAR semantic segmentation. With the advent of sparse convolution [6, 28], more methods have

---

*Equal contribution

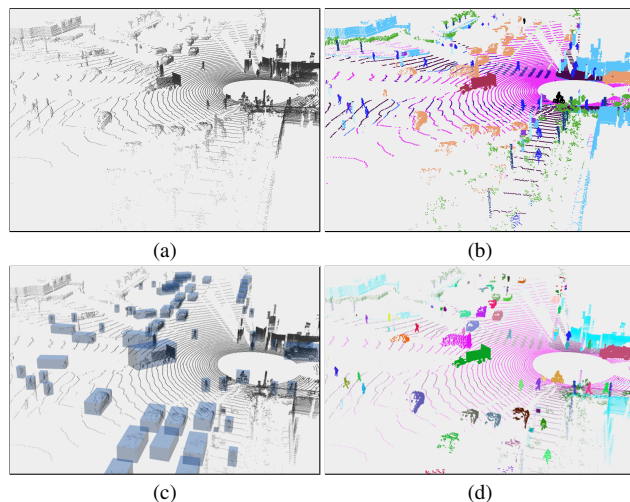†Work done during an internship at TuSimple



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.

started to segment the point clouds in the 3D voxel space. However, due to the requirement of sparse convolution and the need for an encoder-decoder structure for the segmentation task, previous voxel-based LiDAR segmentation networks [5, 36] have difficulty to learn global contextual information. On the other hand, more recent works try to fuse features from multiple views that contain both voxel-level and point-level information. These approaches focus more on exploiting local point geometrical relations to recover fine-grained details for segmentation.

Major LiDAR perception tasks including 3D segmentation (*e.g.*, PolarNet [34]), 3D object detection (*e.g.*, CenterPoint [31]) and panoptic segmentation (*e.g.*, Panoptic-PolarNet [35]), are usually performed in separate and independent networks. In this work, we propose to unify these three major LiDAR perception tasks in a single network that

exploits the synergy between these tasks and achieves state-of-the-art performance, as shown in Figure 1.

Our main contributions are summarized as follows:

- We present an efficient voxel-based LiDAR multi-task network that unifies the major LiDAR perception tasks.

- We propose a novel Global Context Pooling (GCP) module to improve the global feature learning in the encoder-decoder architecture based on 3D sparse convolution.

- We introduce a second-stage refinement module to refine the first-stage semantic segmentation or produce accurate panoptic segmentation results.

## 2. Related Work

**LiDAR Semantic Segmentation** LiDAR point cloud semantic segmentation usually requires transforming the large-scale sparse point cloud into either a 3D voxel map, 2D Bird's Eye View (BEV), or range-view map. Following the trend in the LiDAR point cloud detection methods [14, 29], PolarNet [34] projected the point cloud into a 2D polar BEV map to balance the point distribution in the voxelization procedure. Thanks to the development of the 3D sparse convolution layer [6, 28] in point cloud processing, more methods [5, 36] start using 3D sparse CNN on a voxel feature map for the LiDAR point cloud segmentation. In contrast to the recent methods [5, 27, 30, 36] that focus on the fine-grained features for details, our method aims to enhance the global feature learning in the voxel-based segmentation network.

**Two-stage Semantic Segmentation** Two-stage or multi-stage refinement is a common procedure in point cloud object detection networks [16, 23, 24, 31], but is rarely used in point cloud semantic segmentation. In the image domain, various methods [4, 15, 32, 33, 37] use multiple stages to refine the segmentation prediction from coarse to fine.

**LiDAR Panoptic Segmentation** Recent LiDAR panoptic segmentation methods [12, 21, 35] usually derive from the well-studied segmentation networks [5, 34, 36] 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 [27] and detection [31] methods. According to [9], 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** MultiNet [26] is a seminal work of the image-based multi-task learning that unifies object detection and road understanding tasks in a single network. In LiDAR-based perception, LidarMTL [8] proposed a simple and efficient multi-task network based on 3D sparse convolution and deconvolutions for joint object detection and road understanding.

## 3. Method

### 3.1. Voxel-based LiDAR Segmentation

LiDAR point cloud semantic segmentation aims to predict semantic labels $L = \{l_i | l_i \in (1 \ldots K)\}_{i=1}^N$ for all points $P = \{p_i | p_i \in \mathbb{R}^{3+c}\}_{i=1}^N$ in the point cloud, where $N$ denotes the total number of points, $K$ is the number of classes and each point has $(3 + c)$ input features, *i.e.*, the 3D coordinates $(x, y, z)$, intensity of the reflection, LiDAR elongation, timestamp, etc. We want to learn a segmentation model $f$, parameterized by $\theta$, that minimizes the difference between the prediction $L = f(P|\theta)$ and the point-wise semantic annotation $L^{gt} = \{l_i^{gt} | l_i^{gt} \in (1 \ldots K)\}_{i=1}^N$.

**Voxelization** We first transform the point cloud coordinates $(x, y, z)$ into the voxel index $\{v_i = (\lfloor \frac{x_i}{s_x} \rfloor, \lfloor \frac{y_i}{s_y} \rfloor, \lfloor \frac{z_i}{s_z} \rfloor)\}_{i=1}^N$, where $s$ is the voxel size. Then, we use a simple voxel feature encoder, which only contains Multi-Layer Perceptron (MLP) and maxpooling layers to generate the sparse voxel feature representation $\mathcal{V} \in \mathbb{R}^{M \times C}$:

$$\mathcal{V}_j = \max_{v_i = v_j}(\text{MLP}(p_i)), j \in (1 \ldots M) \qquad (1)$$

where $M$ is the number of unique voxel indices. We also generate the ground truth label of each sparse voxel through majority voting: $L_j^v = \arg\max_{v_i = v_j}(l_i^{gt})$.

**Sparse Voxel-based Encoder-decoder Backbone Network** We designed a 3D counterpart of the 2D UNet [22] as the backbone of our segmentation network, where the voxel feature is gradually downsampled to $\frac{1}{8}$ of the original size in the encoder and upsampled back in the decoder. Key indices are shared among the encoder and decoder in the same scale to keep the same sparsity and reduce repetitive calculation of sparse indices. In each decoder block, the voxel features from previous block and lateral skip-connected features from the encoder are concatenated and further processed by a residual submanifold convolution block and then upsampled by inverse sparse convolution. We supervise our model with the voxel-level label $L^v$ during training and project the predicted label back to the point-level via a de-voxelization step during inference.

### 3.2. LidarMultiNet

The main architecture of LidarMultiNet is illustrated in Figure 2. The main network consists of an encoder-decoder
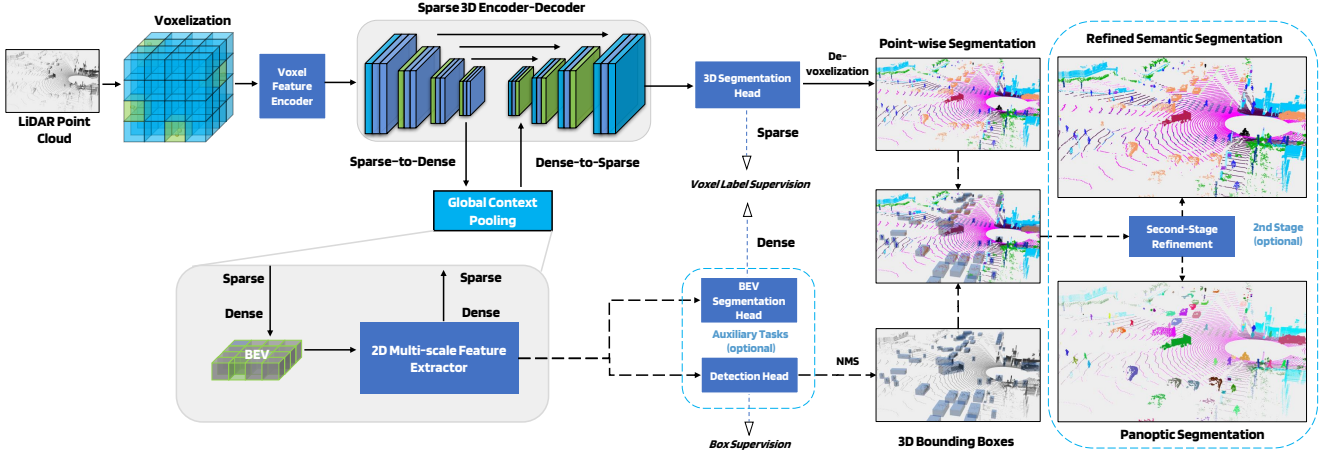
Figure 2. **Main Architecture of the LidarMultiNet**. At the core of our network is a 3D encoder-decoder based on the 3D sparse convolution and deconvolutions. In between the encoder and the decoder, we apply our novel Global Context Pooling (GCP) module to extract contextual information through the conversion between sparse and dense feature maps and via a 2D multi-scale feature extractor. The 3D segmentation head is attached to the decoder output and its predicted voxel labels are projected back to the point-level via a de-voxelization step. Two auxiliary heads, namely the BEV segmentation head and 3D detection head can be attached to the 2D BEV feature map. An optional 2nd-stage produces refined semantic segmentation and panoptic segmentation results.
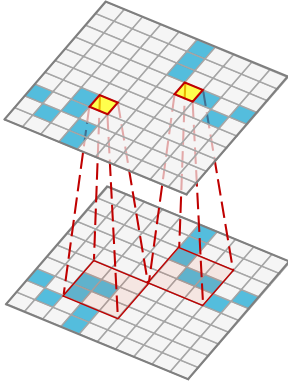


Figure 3. **Submanifold Sparse Convolution.** Convolution only takes place at valid voxels. Features cannot reach the voxels that have big gaps in between.

architecture based on 3D sparse convolution and deconvolutions. The novel GCP module is applied in between the encoder and the decoder. Two auxiliary tasks, namely 3D object detection and BEV segmentation can be attached to the network. An optional second stage can be applied to refine semantic segmentation and generate panoptic segmentation results.

### 3.3. Global Context Pooling

3D sparse convolution drastically reduces the memory consumption and runtime of the 3D CNN, 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 using only submanifold convolution [10] (Figure 3) in the same scale. However, submanifold convolution cannot broadcast features to isolated voxels through stacking multiple convolution layers, limiting the ability to learn global contextual information. Global Context Pooling (GCP) is designed to extract global contextual feature from 3D sparse tensor by projecting it to 2D dense BEV feature map and then converting the dense feature map back after applying 2D multi-scale feature extractor.

As illustrated in Figure 4, given the 3D sparse tensor output by the encoder, we first project it into a dense BEV feature map $\mathcal{F}^{sparse} \in \mathbb{R}^{C \times M'} \to \mathcal{F}^{dense} \in \mathbb{R}^{C \times \frac{D}{d_z} \times \frac{H}{d_x} \times \frac{W}{d_y}}$, where $d$ is the downsampling ratio and $M'$ is the number of valid voxels in the last scale. We concatenate the features across the height dimension together to form a 2D BEV feature map $\mathcal{F}^{bev} \in \mathbb{R}^{(C*\frac{D}{d_z}) \times \frac{H}{d_x} \times \frac{W}{d_y}}$. Then, we use a 2D multi-scale CNN to further extract contextual information. Lastly, we reshape the encoded BEV feature representation to the dense voxel map, then transform it back 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 (*e.g.*, object detection) simply by attaching additional heads with slight increase of computational cost. By utilizing the BEV-level
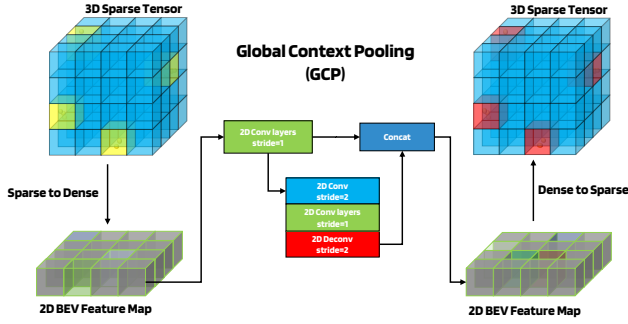
Figure 4. **Illustration of the Global Context Pooling (GCP) module**. 3D sparse tensor is projected to a 2D BEV feature map. Two levels of 2D BEV feature maps are concatenated and then converted back to a 3D sparse tensor.

training like object detection, GCP can enhance the segmentation performance furthermore.

### 3.4. LiDAR Multi-task Learning

**Auxiliary Tasks** We attach the detection head of Center-Point [31], which is widely used in anchor-free 3D detection, to the multi-scale feature extractor. The motivations are twofold. First, the detection loss can serve as the auxiliary loss and help exploit the synergy between the segmentation and detection tasks, since Waymo Open Dataset (WOD) has both segmentation and bounding box labels for the thing classes (*i.e.*, countable objects such as pedestrians and vehicles). Second, the network transcends into a multi-task network that performs both tasks simultaneously.

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 another auxiliary loss during training.

**Multi-task Training and Losses** The 3D segmentation branch predicts voxel-level label $L^v = \{l_j | l_j \in (1 \dots K)\}_{j=1}^M$ given the learned voxel features $\mathcal{F}^{sparse} \in \mathbb{R}^{C_{voxel} \times M}$ output by the 3D decoder. We supervise it through a combination of cross-entropy loss and Lovasz loss [2]: $\mathcal{L}_{SEG} = \mathcal{L}_{ce}^v + \mathcal{L}_{Lovasz}^v$. Note that $\mathcal{L}_{SEG}$ is of sparse loss and the computational cost as well as GPU memory usage is much less than dense loss.

The detection head is applied on the 2D BEV feature map: $\mathcal{F}^{bev} \in \mathbb{R}^{C_{bev} \times \frac{H}{d_x} \times \frac{W}{d_y}}$. It predicts a class-specific heatmap, object size and orientation, and IoU rectification score, which are supervised by the focal loss [17] ($\mathcal{L}_{hm}$) 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 set to $[1, 2, 1]$.

The BEV segmentation head is supervised with $\mathcal{L}_{BEV}$, a dense loss consisting of cross-entropy loss and Lovasz loss:

$$\mathcal{L}_{BEV} = \mathcal{L}_{ce}^{bev} + \mathcal{L}_{Lovasz}^{bev}.$$

Our network is trained end-to-end with supervision by $\mathcal{L}_{SEG}$, $\mathcal{L}_{DET}$ and $\mathcal{L}_{BEV}$, in which both $\mathcal{L}_{DET}$ and $\mathcal{L}_{BEV}$ serve as auxiliary losses during training. We apply a weighted multi-task loss [13] $\mathcal{L}_{total}$ defined as

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

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

### 3.5. Second-stage Refinement

During training the detection serves as an auxiliary loss of the segmentation network, not contributing to the segmentation directly. As shown in Figure 6, the points within a detection bounding box might be misclassified as multiple classes due to the lack of spatial prior knowledge. In order to improve the spatial consistency for the thing classes, we propose a novel point-based approach as the second stage which can also be used to provide accurate panoptic segmentation results. The second stage is illustrated in Figure 5.

Specifically, given all points $P = \{p_i | p_i \in \mathbb{R}^{3+c}\}_{i=1}^N$ in the point cloud, the $B$ predicted bounding boxes, sparse voxel features of the final segmentation branch $\mathcal{F}^{sparse}$, and BEV feature maps $\mathcal{F}^{bev}$, our 2nd stage predicts box-wise class scores $S_{box}$ and point-wise mask scores $S_{point}$. We first transform each point within bounding boxes into its local coordinate as $P_{local}$. Then we devoxelize all points to find their voxel features and process 2nd stage points-wise features as $P_{2nd} = concat(P_{local}, P_{voxel})$. We extract 2nd stage box-wise features $\mathcal{B}_{2nd}$ as in [31] from BEV feature maps $\mathcal{F}^{bev}$. For box-wise aggregation, we generate point-box index $I = \{ind_i | ind_i \in \mathbb{I}, 0 \le ind_i \le B\}_{i=1}^N$. Note that stuff classes points are assigned with index $\varnothing$ and they will not be refined in the second stage. Next, our point-wise backbone takes $P_{2nd}$, $I$ and $\mathcal{B}_{2nd}$ as inputs and outputs point-wise mask scores $S_{point} = \{sp_i | sp_i \in (0, 1)\}_{i=1}^N$ and box-wise classification score $S_{box} = \{sb_i | sb_i \in (0, 1)^{cls+1}\}_{i=1}^B$, where $cls$ denotes the number of thing classes and the one additional class is the unrefined class $\emptyset$. During training, we supervise box-wise class scores through cross-entropy loss and point-wise mask scores through binary cross-entropy loss.

We compute the final scores by fusing the 1st-stage prediction and the 2nd-stage prediction. Moreover, we can integrate the refined segmentation results with the predicted box scores to perform panoptic segmentation.
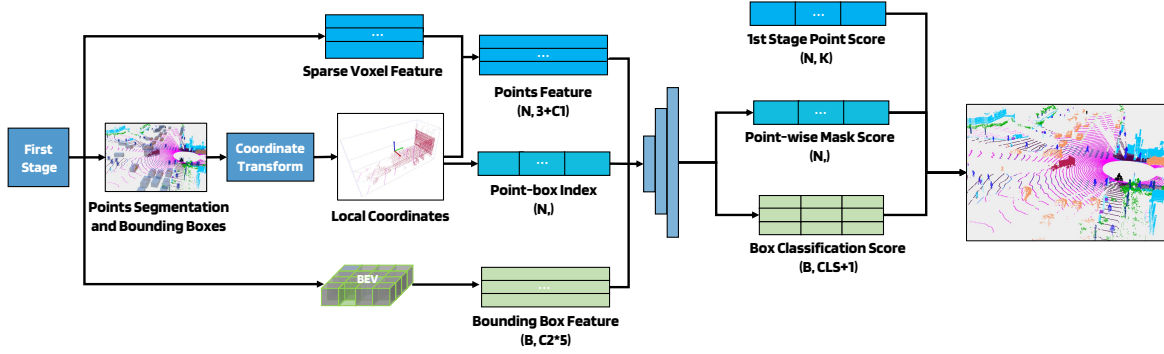
Figure 5. **Illustration of the second-stage refinement pipeline.** The architecture of the second-stage refinement is point-based. We first fuse the detected boxes, voxel-wise features, and BEV features from the 1st stage to generate the inputs for the 2nd stage. We apply the local coordinate transformation to the points within each box. Then, a point-based backbone with MLPs, attention modules, and aggregation modules infers the box-wise classification scores and point-wise mask scores. The final scores are computed by fusing the 1st and 2nd stage predictions.
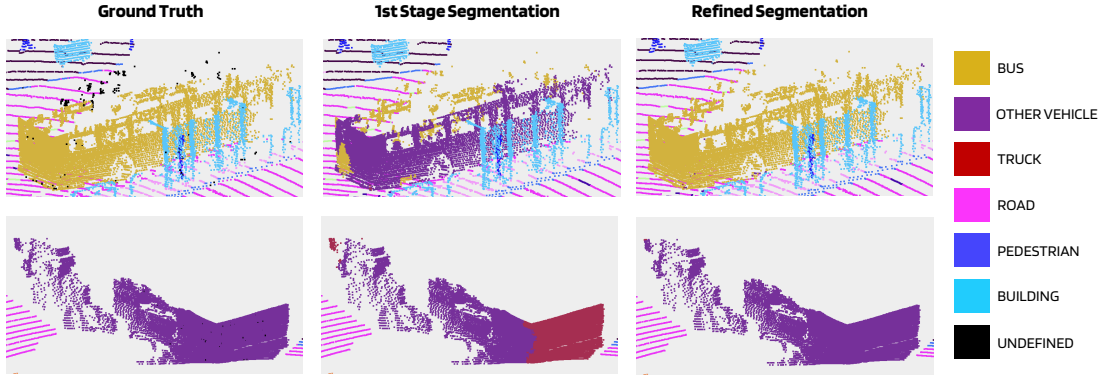


Figure 6. **Examples of the 2nd-stage refinement**. The segmentation consistency of points of the thing objects can be improved by the 2nd stage.

## 4. Experiments

### 4.1. Waymo Open Dataset

The Waymo Open Dataset [25] 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 and the 3D Semantic Segmentation labels are provided for sampled frames. 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. Waymo Open Dataset has semantic labels for a total of 23 classes, including an undefined class.

### 4.2. Implementation Details

As summarized in Table 1, the 3D encoder in our network consists of 4 stages of 3D sparse convolutions with increasing channel width 32, 64, 128, 256. Each stage starts

| Hyperparameters | Module | Values |
|---|---|---|
| #Input features | VFE | 16 |
| Voxel size | Voxelization | $(0.1m, 0.1m, 0.15m)$ |
| Range $x/y$ | Voxelization | $[-75.2m, 75.2m]$ |
| Range $z$ | Voxelization | $[-2m, 4m]$ |
| Downsampling | Encoder | 8 |
| 3D Depth | Encoder | 2, 3, 3, 3 |
| 3D Width | Encoder | 32, 64, 128, 256 |
| 3D Width | Decoder | 128, 64, 32, 32 |
| 2D Depth | GCP | 6, 6 |
| 2D Width | GCP | 128, 256 |

Table 1. Implementation details of the LidarMultiNet

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 en-

| Waymo Leaderboard | mIoU | CAR | TRUCK | BUS | OTHERVEHICLE | MOTORCYCLIST | BICYCLIST | PEDESTRIAN | SIGN | TRAFFICLIGHT | POLE | CONE | BICYCLE | MOTORCYCLE | BUILDING | VEGETATION | TREETRUNK | CURB | ROAD | LANEMARKER | OTHERGROUND | WALKABLE | SIDEWALK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **LidarMultiNet** | **71.13** | **95.86** | **70.57** | 81.44 | 35.49 | 0.77 | **90.66** | **93.23** | **73.84** | **33.14** | **81.37** | 64.68 | 69.92 | 76.73 | **97.37** | **88.92** | **73.58** | 76.54 | 93.21 | **50.37** | **53.82** | **75.48** | **87.95** |
| SegNet3DV2 | 70.48 | 95.73 | 69.03 | 79.74 | **37.0** | 0 | 88.77 | 92.66 | 71.82 | 30.02 | 80.85 | **65.97** | 69.53 | **76.97** | 97.15 | 88.18 | 72.76 | 76.4 | **93.27** | 49.49 | 52.61 | 75.4 | 87.25 |
| HorizonSegExpert | 69.44 | 95.55 | 68.93 | **84.38** | 29.39 | 0.95 | 86.06 | 92.22 | 72.71 | 32.27 | 80.56 | 55.67 | 68.56 | 71.7 | 97.18 | 87.81 | 72.49 | 76.0 | 93.09 | 48.91 | 52.21 | 73.78 | 87.29 |
| HRI_HZ_SMRPV | 69.38 | 95.79 | 66.97 | 78.36 | 32.8 | 0.04 | 89.06 | 91.83 | 72.93 | 31.73 | 79.82 | 61.33 | **70.23** | 76.02 | 96.94 | 87.2 | 72.15 | 73.8 | 92.6 | 48.02 | 50.82 | 71.94 | 86.03 |
| Waymo_3DSEG | 68.99 | 95.47 | 69.54 | 77.85 | 31.29 | 0.67 | 84.4 | 91.12 | 72.73 | 31.45 | 80.12 | 61.25 | 68.87 | 74.87 | 96.96 | 87.89 | 72.93 | 74.62 | 92.61 | 44.68 | 49.33 | 73.1 | 86.13 |
| SPVCNN++ | 67.7 | 95.12 | 67.73 | 75.61 | 35.16 | 0 | 85.1 | 91.57 | 73.24 | 31.94 | 78.85 | 61.4 | 65.97 | 73.6 | 90.82 | 86.58 | 70.54 | 75.53 | 91.73 | 41.31 | 40.22 | 71.5 | 85.87 |
| PolarFuse | 67.28 | 95.06 | 67.79 | 77.04 | 30.54 | 0 | 79.08 | 89.61 | 65.37 | 30.38 | 78.53 | 57.3 | 62.97 | 67.96 | 96.38 | 87.07 | 70.26 | 75.1 | 92.54 | 48.91 | 51.75 | 71.39 | 85.02 |
| LeapNet | 66.89 | 94.45 | 65.68 | 79.01 | 30.74 | 0.01 | 79.44 | 90.07 | 70.46 | 30.34 | 77.67 | 52.71 | 61.58 | 66.84 | 96.65 | 87.01 | 69.8 | 74.49 | 92.61 | 44.91 | 47.17 | 73.59 | 86.34 |
| 3DSEG | 66.77 | 94.64 | 66.95 | 77.61 | 30.67 | **2.11** | 78.59 | 89.18 | 70.01 | 31.0 | 77.4 | 56.73 | 60.81 | 67.91 | 96.55 | 87.33 | 70.89 | 70.92 | 91.79 | 42.99 | 46.78 | 72.95 | 85.04 |
| CAVPercep | 63.73 | 93.64 | 62.83 | 68.12 | 22.83 | 1.89 | 71.73 | 87.39 | 67.03 | 29.41 | 74.29 | 55.03 | 55.6 | 60.84 | 96.15 | 86.48 | 68.06 | 68.33 | 91.24 | 41.26 | 46.19 | 70.55 | 83.15 |
| VS-Concord3D | 63.54 | 92.6 | 66.9 | 73.13 | 24.26 | 0.83 | 76.77 | 85.51 | 66.88 | 29.98 | 75.81 | 53.86 | 62.26 | 68.11 | 86.04 | 75.13 | 66.95 | 67.95 | 90.6 | 40.43 | 45.43 | 65.6 | 82.96 |
| RGBV-RP Net | 62.61 | 94.87 | 67.48 | 74.91 | 33.09 | 0 | 77.34 | 88.66 | 68.05 | 28.68 | 74.78 | 37.69 | 53.82 | 64.92 | 96.59 | 86.47 | 67.18 | 70.93 | 90.97 | 23.7 | 24.01 | 68.89 | 84.48 |

Table 2. Waymo Open Dataset Semantic Segmentation Leaderboard. Our **LidarMultiNet** reached the highest mIoU of 71.13 and achieved the best accuracy on 15 out of the 22 classes.

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

Table 3. Ablation studies for 3D semantic segmentation on the WOD validation set. We show the improvement introduced by each component compared to our LidarMultiNet base network.

coder. The 3D decoder has 4 stages of 3D sparse deconvolution blocks with decreasing channel widths 128, 64, 32, 32. 2D dense BEV features are obtained by projecting the 3D sparse tensor via channel concatenation across its height dimension.

During training, the point cloud range is set to $[-75.2m, 75.2m]$ for $x$ axis and $y$ axis, and $[-2m, 4m]$ for $z$ axis, and the voxel size is set to $(0.1m, 0.1m, 0.15m)$. For data augmentation, we use random flipping with respect to $xz$-plane and $yz$-plane, and global scaling with a random factor from $[0.95, 1.05]$. We also apply a random global rotation between $[-\pi/4, \pi/4]$ and a random translation with a normally distributed displacement $\Delta d \sim \mathcal{N}(0, 0.5^2)$ along all axes.

Following [31], 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 input 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.

We train the models using AdamW [18] optimizer with one-cycle learning rate policy [11], with max learning rate 3e-3, weight decay 0.01, and momentum 0.95 to 0.85. We use a batch size of 2 on each of the 8 V100 GPUs. For 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.

### 4.3. Ablation Study

We ablate each component of the LidarMultiNet and the result on the WOD validation set is shown in Table 3. Our base LidarMultiNet reaches a mIoU of 69.90 on the validation set. On top of this baseline, multi-sweep input (*i.e.*, including past two frames) brings a 0.59 mIoU improvement. Our proposed 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 further improves the mIoU by 0.34, forming our best single model on the WOD validation set.

### 4.4. Test-Time Augmentation and Ensemble

In order to further improve the performance, we apply Test-Time Augmentation (TTA) and Ensemble methods. Specifically, we use flipping with respect to $xz$-plane and $yz$-plane, [0.95, 1.05] for global scaling, and $[\pm 22.5°, \pm 45°, \pm 135°, \pm 157.5°, 180°]$ for yaw rotation. Besides, we found pitch and roll rotation were helpful in the segmentation task, and we use $\pm 8°$ for pitch rotation and $\pm 5°$ for roll rotation. In addition, we also apply $\pm 0.2m$ translation along $z$-axis for augmentation.

Besides the best single model, we also explored the network design space and designed multiple variants for model ensemble. For example, more models are trained with smaller voxel size $(0.05m, 0.05m, 0.15m)$, smaller downsample factor $(4\times)$, different channel width (64), without the 2nd stage, or with more past frames (4 sweeps). Table 4 shows the improvement brought by TTA and ensemble on the validation set. For our submission to the leaderboard, a total of 7 models are ensembled to generate the segmentation result on the test set.

### 4.5. Waymo Open Dataset 3D Semantic Segmentation Challenge Leaderboard

For our final submission to the leaderboard, we train our models from scratch on the joint dataset of Waymo training and validation splits. Table 2 is the final WOD semantic

| Methods | mIoU |
|---|---|
| Best single model | 72.40 |
| + TTA | 73.05 |
| + Ensemble | 73.78 |

Table 4. Improvements made on top of the best single model on the WOD validation set

segmentation leaderboard and shows that our LidarMulti-Net achieves a mIoU of 71.13 and ranks the 1st place on the leaderboard, and also has the best IoU for 15 out of the total 22 classes.

# 5. Conclusion

Our proposed LidarMultiNet reached the 1st place in the Waymo Open Dataset 3D semantic segmentation challenge 2022. As one of our future works, we plan to verify the effectiveness of LidarMultiNet on 3D object detection and panoptic segmentation benchmarks as well.

# References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 1

[2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018. 4

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1

[4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. In *NeurIPS*, 2018. 2

[5] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *CVPR*, 2021. 1, 2

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *ICCV*, 2019. 1, 2

[7] Waymo Open Dataset. Waymo open dataset challenges - 3d semantic segmentation leaderboard https://waymo. com/open/challenges/2022/3d-semantic-segmentation/, 2022. 1

[8] Di Feng, Yiyang Zhou, Chenfeng Xu, Masayoshi Tomizuka, and Wei Zhan. A simple and efficient multi-task network for 3d object detection and road understanding. In *IROS*, 2021. 2

[9] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. In *arXiv*, 2021. 2

[10] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 3

[11] Sylvain Gugger. The 1cycle policy https://sgugger. github.io/the-1cycle-policy.html, 2018. 6

[12] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *CVPR*, 2021. 2

[13] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 4

[14] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 2

[15] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, 2016. 2

[16] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *CVPR*, 2021. 2

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 4

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017. 6

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1

[20] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 1

[21] Ryan Razani, Ran Cheng, Enxu Li, Ehsan Taghavi, Yuan Ren, and Liu Bingbing. Gp-s3net: Graph-based panoptic sparse semantic segmentation network. In *ICCV*, 2021. 2

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[23] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *ICCV*, 2021. 2

[24] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020. 2

[25] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1, 5

[26] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *arXiv*, 2016. 2

[27] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *ICCV*, 2021. 2

[28] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. In *Sensors*, 2018. 1, 2

[29] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018. 2

[30] Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Drinet: A dual-representation iterative learning network for point cloud segmentation. In *ICCV*, 2021. 2

[31] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 1, 2, 4, 6

[32] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 2

[33] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *ICCV*, 2019. 2

[34] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, 2020. 1, 2

[35] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *CVPR*, 2021. 1, 2

[36] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021. 1, 2

[37] Zhuotun Zhu, Yingda Xia, Wei Shen, Elliot Fishman, and Alan Yuille. A 3d coarse-to-fine framework for volumetric medical image segmentation. In *3DV*, 2018. 2