LiDARFormer: A Unified Transformer-based Multi-task Network for LiDAR Perception

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

[1]TuSimple
[2]University of Central Florida

Abstract

*There is a recent trend in the LiDAR perception field towards unifying multiple tasks in a single strong network with improved performance, as opposed to using separate networks for each task. In this paper, we introduce a new LiDAR multi-task learning paradigm based on the transformer. The proposed **LiDARFormer** utilizes cross-space global contextual feature information and exploits cross-task synergy to boost the performance of LiDAR perception tasks across multiple large-scale datasets and benchmarks. Our novel transformer-based framework includes a cross-space transformer module that learns attentive features between the 2D dense Bird's Eye View (BEV) and 3D sparse voxel feature maps. Additionally, we propose a transformer decoder for the segmentation task to dynamically adjust the learned features by leveraging the categorical feature representations. Furthermore, we combine the segmentation and detection features in a shared transformer decoder with cross-task attention layers to enhance and integrate the object-level and class-level features. LiDARFormer is evaluated on the large-scale nuScenes and the Waymo Open datasets for both 3D detection and semantic segmentation tasks, and it outperforms all previously published methods on both tasks. Notably, LiDARFormer achieves the state-of-the-art performance of 76.4% L2 mAPH and 74.3% NDS on the challenging Waymo and nuScenes detection benchmarks for a single model LiDAR-only method.*

1. Introduction

LiDAR point cloud detection and semantic segmentation tasks aim to predict the object-level 3D bounding boxes and point-level semantic labels, which are among the most

---

[*]Work done during an internship at TuSimple.
[†]Contributed equally.

Figure 1: **LiDAR Perception Network Designs.** LiDAR detection (a) and segmentation (b) networks typically extract feature representations on distinct feature maps. While a recent multi-task network [66] (c) integrates these tasks into a single network, it often overlooks differences among feature maps and the higher-level connections between tasks. Our network (d) utilizes transformer attention to establish more effectively the transformations between 3D sparse and 2D dense features. Moreover, the cross-task information is further shared through class-level and object-level feature embeddings in the multi-task transformer decoder.

fundamental tasks in autonomous vehicle perception. With the recent release of the large-scale LiDAR point cloud

datasets [3, 46], there has been a surge of interest in integrating these tasks into a single framework. Current methods [19, 66] rely on voxel-based networks with sparse convolution [63, 14] for leading performance. However, different tasks are only connected through sharing the same low-level features without considering the high-level contextual information that is highly related among those tasks. On the other hand, more recent works [44, 48, 60] try to fuse features from multiple views that contain both voxel-level and point-level information. These approaches focus more on exploiting local point geometric relations to recover fine-grained details. The problem of efficiently extracting and sharing global contextual information in LiDAR perception tasks is still by and large underexplored.

Meanwhile, transformer-based network structures [4, 53, 12, 59, 73] start to exhibit an outstanding performance on 2D image detection and segmentation tasks. Apart from directly replacing the conventional CNN with the transformer encoder [16, 71], various methods [4, 80, 12, 28] explore using the transformer decoder to extract objects or class-level feature representations, which are served as strong contextual information for feature learning. This transformer decoder design is then adopted in recent LiDAR perception methods [78, 1, 37]. However, the transformer decoders used for LiDAR detection and segmentation tasks are performed independently on different feature maps and are not yet unified.

Is it possible to develop a unified transformer-based multi-task LiDAR perception network with the ability to learn global context information? To accomplish this goal, we introduce three novel components in a voxel-based framework. The first component is a cross-space transformer module that enhances the feature mapping between the 3D sparse voxel space and the 2D dense BEV space. These two spaces are frequently used to obtain feature representations for segmentation and detection tasks, respectively. Second, we propose a transformer-based refinement module as the segmentation decoder. The module uses a transformer to extract class feature embeddings and refine voxel features through bidirectional cross-attention. Lastly, we propose a multi-task learning structure that combines segmentation and detection transformer decoders into a unified transformer decoder. By doing so, the network can transfer high-level features through cross-task attention, as depicted in Figure 1. These three innovative components result in a powerful network, named **LiDARFormer**, for the next generation of LiDAR perception.

We evaluate our method on two challenging large-scale LiDAR datasets: the nuScenes dataset [3] and the Waymo Open Dataset [46]. Our method sets new state-of-the-art standards both in detection and semantic segmentation, by achieving 74.3% NDS on the nuScenes 3D detection and 81.5% mIoU on the nuScenes semantic segmentation. Li-

DARFormer also achieves 76.4% mAPH in the Waymo Open Dataset detection set, surpassing thus all previous methods.

Our main contributions are summarized as follows:

- We propose a cross-space transformer module to improve feature learning when transferring features between sparse voxel features and dense BEV features in the multi-task network.

- We present the first LiDAR cross-task transformer decoder that bridges the information learned across object-level and class-level feature embedding.

- We introduce a transformer-based coarse-to-fine network that utilizes a transformer decoder to extract class-level global contextual information for the LiDAR semantic segmentation task.

- Our network achieves state-of-the-art 3D detection and semantic segmentation performances on two popular large-scale LiDAR benchmarks.

2. Related Work

Voxel-based LiDAR Point Cloud Perception Unlike most point cloud networks [41, 42, 30, 58, 22, 61, 49, 75] that directly learn point-level features in outdoor or indoor point cloud data, LiDAR point cloud perception usually requires transforming the large-scale sparse point cloud into either a 3D voxel map [77, 81], 2D BEV [64, 27, 74], or range-view map [47, 18, 56, 57, 38, 15]. Thanks to the development of the 3D sparse convolution layer [63, 14] in point cloud processing, voxel-based methods are becoming dominant in terms of both high performance and efficient runtime. CenterPoint [69] and AFDet [20] adopted the anchor-free design that detects objects through heatmap classification. Cylinder3D [81] utilized the cylindrical voxel partition to extract the voxel-level features. LargeKernel3D [10] showed that the long-range information from a bigger receptive field can significantly improve the performance. LidarMultiNet [66] presented a multi-task learning network that unifies different LiDAR perception tasks.

Voxel-based methods have to make a trade-off between accuracy and complexity due to the information loss introduced during the projection or voxelization. To alleviate the quantization error, some recent methods [48, 44, 68, 60] propose to fuse features from multi-view feature maps, combining point-level information with 2D BEV/range-view and 3D voxel features. PVRCNN [44] and SPV-NAS [48] used two concurrent point-level and voxel-level feature encoding branches, where these two features were connected at each network block. RPVNet [60] further combined all point, voxel, and range image features in an

Figure 2: **The architecture of LiDARFormer.** Our network first transforms the point cloud into a sparse voxel map. Next, sparse 3D CNN is used to extract voxel feature representation. Between the encoder and the decoder, we use a Cross-space Transformer (**XSF**) module to learn long-range information in the BEV map. Additionally, we use a cross-task transformer decoder (**XTF**) to extract class-level and object-level feature representations, which are fed into task-specific heads to generate the detection and segmentation predictions.

encoder-decoder segmentation network through a gated fusion module. In contrast to these methods that focus on fine-grained features for details, our method aims to enhance global feature learning in the voxel-based network.

Segmentation Refinement In the image domain, various methods [29, 82, 8, 72, 70] use multiple stages to refine the segmentation prediction from coarse to fine. ACFNet [72] proposed an attentional class feature module to refine the pixel-wise features based on a coarse segmentation map. OCR [70] further advanced the idea to use a bidirectional connection between pixel-wise features and object-contextual representations to enrich the features. In comparison, refinement modules have been rarely used in point cloud semantic segmentation.

Transformer Decoder Transformer [50] structure has gained huge popularity in recent years. Built on the development of 2D transformer backbones [16, 71], various methods [80, 76, 53, 12, 59, 73] are proposed to tackle the 2D detection and segmentation problems. Depending on the source of the input, the vision transformers can be categorized into encoder [71, 76, 59] and decoder [4, 53, 12, 70, 73, 28]. A transformer encoder usually serves as a feature encoding network to replace the conventional neural networks, while a transformer decoder is used to extract class-level or instance-level feature representations for the downstream tasks. In the LiDAR domain, several detection methods [39, 65, 36, 43, 1, 40, 32, 78] have started to integrate the transformer decoder structure into the previous frameworks. Besides the performance improvement, the transformer decoder demonstrates great potential for an end-to-end training [39] and multi-frame [65, 78] / modality [1, 32] feature fusion. However, studying effective meth-

ods of using a transformer decoder in LiDAR segmentation is still an underexplored area. In this paper, we propose a novel class-aware global contextual refinement module for LiDAR segmentation based on the transformer decoder, while exploiting the synergy between detection and segmentation decoders.

3. Method

In this section, we present the design of LiDARFormer. As shown in Figure 2, our framework consists of three parts: (3.1) A 3D encoder-decoder backbone network using 3D sparse convolution; (3.2) A Cross-space Transformer (XSF) module extracting large-scale and context features in the BEV; (3.3) A Cross-task Transformer (XTF) decoder that aggregates class-wise and object-wise global contextual information from voxel and BEV feature maps. Our network adopts the multi-task learning framework from LidarMulti-Net [66], but further associates the global features between segmentation and detection through a shared cross-task attention layer.

3.1. Voxel-based LiDAR Perception

LiDAR point cloud semantic segmentation and object detection aim to predict pixel-wise semantic labels $L = \{l_i | l_i \in (1 \dots K)\}_{i=1}^{N}$ and object bounding boxes $O = \{o_i | o_i \in \mathbb{R}^7\}_{i=1}^{B}$ in a point cloud $P = \{p_i | p_i \in \mathbb{R}^{3+c}\}_{i=1}^{N}$, where N denotes the number of points, B and K are the number of objects and classes. Each point has $(3 + c)$ input features, i.e. the 3D coordinates (x, y, z), the intensity of the reflection, LiDAR elongation, timestamp, etc. Each object is represented by its 3D location, size and orientation.

Voxelization We first transform the point cloud

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

$$\mathcal{V}_j = \max_{\mathcal{I}_i = \mathcal{I}_j} (\text{MLP}(p_i)), j \in (1 \dots M) \quad (1)$$

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

Sparse Voxel-based Backbone Network We use a VoxelNet [77] as the backbone of our network, where the voxel features are gradually downsampled to $\frac{1}{8}$ of the original size in the encoder. The sparse voxel features are projected onto the dense BEV map, followed by a 2D multi-scale feature extractor to extract the global information. For the detection task, we attach a detection head to the BEV feature map to predict the object bounding boxes. For the segmentation task, the BEV feature is reprojected to the voxel space, where we use a U-Net decoder to upsample the feature map back to the original scale. We supervise our model with the voxel-level label L^v and project the predicted label back to the point level via a de-voxelization step during inference.

3.2. Cross-space Transformer

As shown in Figure 1, voxel-based LiDAR detection and segmentation generally require the backbone network to extract feature representations on the 2D dense BEV space and 3D sparse voxel space, respectively. To overcome the challenge of merging the features learned from these two tasks, the previous multi-task network [66] proposed a global context pooling module to directly map the features based on their location without considering differences in sparsity. In contrast, we propose a cross-space Transformer module that utilizes deformable attention to enhance feature extraction between these spaces to further increase the receptive field.

As shown in Figure 2, we employ a cross-space Transformer to 1) convert the sparse voxel features in the last scale $\mathcal{F}_{in}^{sparse} \in \mathbb{R}^{C \times M'}$ into dense BEV features (*Sparse-to-dense*), and 2) convert the dense BEV features from 2D multi-scale feature extractor $\mathcal{F}^{dense} \in \mathbb{R}^{(C \times \frac{D}{d_z}) \times \frac{H}{d_x} \times \frac{W}{d_y}}$ to sparse voxel features $\mathcal{F}_{out}^{sparse} \in \mathbb{R}^{C \times M'}$, where d is the downsampling ratio and M' is the number of valid voxels in the encoder's last scale (*Dense-to-sparse*). The cross-space Transformer is illustrated in Figure 3. Specifically, in Figure 3a, $\mathcal{F}^{dense}$ is divided into slices by height as $\mathcal{F}_{3D}^{dense} \in \mathbb{R}^{C \times \frac{D}{d_z} \times \frac{H}{d_x} \times \frac{W}{d_y}}$. Then we take the features from $\mathcal{F}_{3D}^{dense}$ at the valid coordinates (u, v, h) of $\mathcal{F}_{in}^{sparse}$ as query $\mathbf{Q}_{3D}$ to predict $\mathcal{F}_{out}^{sparse}$. The deformable attention [80] is adopted as a self-attention layer to explore global informa-

(a) Dense-to-sparse Cross-space Transformer

(b) Sparse-to-dense Cross-space Transformer

Figure 3: **Illustration of the Cross-space Transformer (XSF) module.** XSF consists of two parts: a multi-height deformable self-attention, and a feed-forward network. (a) convert dense BEV features to sparse voxel features, (b) convert sparse voxel features to dense BEV features with two more densify operations.

tion in the dense feature map. Since $\mathcal{F}^{dense}$ lacks height information, due to the fact that 2D multi-scale feature extractor mainly focuses on BEV-level information, we develop a multi-head multi-height attention module to learn features along all heights: For every reference voxel whose location is $\xi = (u, v)$ on the sliced BEV feature map at height h, the deformable self-attention uses a linear layer to learn BEV offsets $\Delta\xi$ at all heads and heights. The features at $\xi + \Delta\xi$ will be sampled from different multi-heights-sliced BEV feature maps through bilinear interpolation. The output of the multi-height deformable self-attention $\chi(p)$ can be formulated as:

$$\chi(p) = \sum_{i=1}^{N_{head}} W_i [\sum_{j=1}^{N_{height}} \sum_{r=1}^{R} \sigma(W_{ijr} q_p) W_i' x^j (\xi + \Delta\xi_{ijr})] \quad (2)$$

where N_{head} is the number of heads, $N_{height} = \frac{D}{d_z}$ is the number of heights, W is learnable weights, R is the number of sampling points, x^j is the multi-heights-sliced BEV features, q_p is the query features at the position ξ, and $\sigma(W_{ijr} q_p)$ is the attention weight.

Since the Dense-to-sparse cross-space Transformer is applied after the 2D feature extractor, it will not affect the learned 2D BEV features, thus has limited impact on increasing the detection performance. To increase the receptive field of the 2D BEV feature extractor, we add a cross-space Transformer module converting $\mathcal{F}_{in}^{sparse}$ into dense

BEV features in a similar manner, as shown in Figure 3b. It equips the BEV feature which will be fed into a 2D multi-scale feature extractor with more context information.

3.3. Cross-task Transformer Decoder

Although object detection and semantic segmentation share correlated information, they are usually learned in two separate network structures. LidarMultiNet [66] demonstrates that through sharing intermediate feature representation, both detection and segmentation performance can get improved. However, no high-level information is shared during the training of the multi-task network. To further explore the multi-task learning synergy, we propose to use a shared transformer decoder to bridge between the class-level information from segmentation and the object-level information from detection. In this section, we first present a novel segmentation decoder that uses class feature embedding to perform dynamic segmentation. Then, we introduce an approach to connect this segmentation decoder with the conventional detection decoder through cross-task attention.

Segmentation Transformer Decoder Inspired by the coarse-to-fine methods [72, 70] in the 2D image segmentation, we propose a class-aware feature refinement module to enhance the global information learning for the segmentation task. We use an initial segmentation prediction to generate the class feature embedding. Then, we use a transformer with bidirectional cross-attention to refine both voxel and class feature representations. The class feature representation is also served as the dynamic kernel in the later segmentation head.

Given an initial semantic segmentation score $y = \{pred_j | pred_j \in [0,1]^K\}_{j=1}^M$, and its encoded feature representation $\mathcal{F} \in \mathbb{R}^{M \times C}$, where M is the number of valid predictions, we generate the class feature embedding $\varepsilon = \{\varepsilon_k | k \in \{1 \ldots K\}\}$ as follows: $\varepsilon_k = \frac{\sum_{j=1}^M pred_j[k] \cdot \mathcal{F}_j}{\sum_{j=1}^M pred_j[k]}$. In our cross-task transformer, we use a coarse prediction and its corresponding BEV features to initialize the class feature embedding. The class feature embedding ε encapsulates the class center information based on the coarse segmentation result of each scan. Assuming that points from the same class have similar or correlated features in the encoded feature embedding, the learned class features can help the network distinguish the edge points that are ambiguous in the segmentation head.

Similar to [70], we propose to use a transformer decoder to further extract the class feature embedding and refine the original voxel features simultaneously through bidirectional cross-attention. As shown in Figure 4, our transformer structure has two parallel branches for the voxel feature $\mathcal{V} \in \mathbb{R}^{M \times C}$ and the class feature $\varepsilon \in \mathbb{R}^{K \times C}$.

We use a standard transformer decoder [50], containing a

Figure 4: **Cross-task Transformer (XTF).** The segmentation and detection decoders share a self-attention layer to transfer the cross-task features. In the segmentation decoder, we use a bidirectional cross-attention to refine voxel features based on the aggregated class feature embedding. For simplicity, the skip connection and the layer norm are ignored in this figure.

multi-head self-attention layer, a multi-head cross-attention layer, and a feed-forward layer, to extract class features using ε as the initial query embedding. In the cross-attention layer, query $\mathbf{Q}_c$ is the linear projection of ε, while key $\mathbf{K}_v$ and value $\mathbf{V}_v$ are the linear projection of $\mathcal{V}$. It can be formulated as :

$$\text{CrossAtt}(\mathcal{V} \to \varepsilon) = \text{Softmax}(\frac{\mathbf{Q}_c \mathbf{K}_v^T}{\sqrt{C}})\mathbf{V}_v. \qquad (3)$$

Next, we use an inverse transformer decoder to transfer the encoded class features back to the voxel features. It is infeasible to use self-attention in the voxel branch due to the huge size of the voxels. Conversely, query $\mathbf{Q}_v$ is from the linear projection of $\mathcal{V}$, key $\mathbf{K}_c$, and value $\mathbf{V}_c$ are the linear projection of the output ε' in the class branch:

$$\text{CrossAtt}(\varepsilon' \to \mathcal{V}) = \text{Softmax}(\frac{\mathbf{Q}_v \mathbf{K}_c^T}{\sqrt{C}})\mathbf{V}_c. \qquad (4)$$

The output voxel feature $\mathcal{V}'$ is then concatenated to the original features $\mathcal{V}^r = (\mathcal{V}, \mathcal{V}')$ for the segmentation head.

Dynamic Kernel Conventional segmentation networks use a segmentation head that consists of convolution or linear layers to reduce the channel size of a voxel feature to the number of classes to make the prediction. The weights learned in the segmentation head are shared among different frames. Therefore the segmentation head is hard to adjust to the varying conditions of scenes. Following the new trend in the image instance segmentation [54, 53, 12, 28], we directly use the learned class feature embedding ε' as

the kernel to generate the semantic logits $\mathcal{S} = \frac{\Phi(\mathcal{V}^r) \cdot \varepsilon'^T}{\sqrt{C}} \in \mathbb{R}^{M \times K}$, where Φ is the convolution layer that reduces the channel size of the voxel feature to C.

Cross-task Attention As shown in Figure 4, we adopt the detection transformer decoder from the well-studied CenterFormer [78], which represents the object-level feature as center query embedding initialized from BEV center proposals. We initialize the class feature embedding ε using the BEV feature. Class and center features are concatenated and then sent into a shared transformer decoder, where the information between detection and segmentation tasks are transferred to each other through a cross-task self-attention layer. Due to the memory limitation, the class and center feature aggregate features separately from the voxel and BEV feature maps, respectively.

4. Experiments

In this section, we present the experimental results of our proposed method on two large-scale public LiDAR point cloud datasets: the nuScenes dataset [3] and the Waymo Open Dataset[46], both of which have 3D object bounding boxes and pixel-wise semantic label annotations. We also provide a detailed ablation study of the improvements and in-depth analysis of our model. More details and visualization are included in the supplementary materials.

4.1. Datasets

The **NuScenes** dataset is a large-scale autonomous driving dataset developed by Motional. It contains 1000 scenes of 20s video data, each of them captured by a 20Hz Velodyne HDL-32E Lidar sensor with 32 vertical beams. NuScenes provides annotations of object bounding boxes and pixel-wise semantic labels at each keyframe sampled at 2 Hz. 16 classes are used for the semantic segmentation evaluation. 10 foreground object ("thing") classes with ground truth bounding box labels are used for the object detection task. For the nuScenes detection task, mean Average Precision (mAP) and NuScenes Detection Score (NDS) are used as the metrics. For semantic segmentation, mean Intersection over Union (mIoU) is used as the metric.

The **Waymo Open Dataset (WOD)** contains around 2000 scenes of 20s video data that is collected at 10Hz by a 64-line LiDAR sensor. Even though WOD provides object bounding box annotation for every frame, it only has the semantic annotation in some key frames sampled at 2Hz. WOD has semantic labels for 23 classes and uses the standard mIoU as the evaluation metric. For the object bounding box annotation, 3 classes of vehicles, pedestrians, and cyclists are calculated into the 3D detection metrics. Average Precision Weighted by Heading (APH) is used as the main detection evaluation metric. The ground truth objects are categorized into two levels of difficulty, LEVEL_1 (L1)

is assigned to the examples that have more than 5 LiDAR points and not in the L2 category, while LEVEL_2 (L2) is assigned to examples that have at least 1 LiDAR point and at most 5 points or are manually labeled as hard. The primary metric mAPH L2 is computed by considering both L1 and L2 examples.

4.2. Experiment Setup

We used the AdamW optimizer with the one-cycle scheduler to train our model for 20 epochs. Most experiments are conducted on 8 Nvidia A100 GPUs with batch size 16. For the multi-task training experiments on WOD, we used batch size 8 because of the GPU memory limits. We used the voxel size of $[0.1, 0.1, 0.2]$ for nuScenes datasets, and $[0.1, 0.1, 0.15]$ for Waymo Open Dataset. For the segmentation task, we used a combination of cross-entropy loss and Lovasz loss [2] to optimize our network. For the detection task, we followed [69] to use the common center heatmap classification loss and bounding box regression loss. We added an auxiliary loss on the output voxel features or BEV features to supervise the segmentation prediction, which is used to initialize the class feature embedding. All losses are fused by multi-task uncertainty weighting strategy [25]. We concatenated the points from the previous 9 scans to the current point cloud in nuScenes, and 2 scans in WOD. Standard data augmentation strategy [51, 66] were applied when training the model. More network and training details are included in the supplementary materials.

4.3. Main Results

We present the detection and segmentation benchmark results on both nuScenes and WOD. All results of other methods in the test set are from the literature, where most of them apply test-time augmentation (TTA) or an ensemble method to increase the performance. In addition to our multi-task network, we also provide the results of the segmentation-only variation of our model, which is trained only with the segmentation transformer decoder.

NuScenes In Table 1 and Table 2, we compare LiDAR-Former with other state-of-the-art methods on the test set of nuScenes. LiDARFormer reaches the top performance of 81.5% mIoU, 71.5% mAP, and 74.3% NDS for a single model result. Notably, the results of the detection task outperform all previous methods by a large margin, especially for the mAP metric. Although the segmentation performance of LiDARFormer is only 0.1% higher than Lidar-MultiNet, LiDARFormer does not require a second stage and can be trained end-to-end by comparison. To fairly compare with other methods without the effect of test-time augmentation, we also demonstrate the performance on the validation set of nuScenes in Table 3. Our segmentation-only LiDARFormer achieves a 81.7% mIoU performance

Table 1: Detection results on the `test` split of nuScenes. "TTA" means test-time augmentation.

Model	Ref	mAP	NDS
CBGS [79]	arXiv 2019	52.8	63.3
CenterPoint [69]	CVPR 2021	58.0	65.5
HotSpotNet [5]	ECCV 2020	59.3	66.0
Object DGCNN [55]	NeurIPS 2021	58.7	66.1
AFDetV2 [23]	AAAI 2022	62.4	68.5
Focals Conv [9]	CVPR 2022	63.8	70.0
TransFusion-L [1]	CVPR 2022	65.5	70.2
LargeKernel3D [10]	CVPR 2023	65.3	70.5
SphereFormer [66]	CVPR 2023	65.5	70.7
LidarMultiNet [66]	AAAI 2023	67.0	71.6
MDRNet-TTA [24]	arXiv 2022	67.2	72.0
LargeKernel3D-TTA [10]	CVPR 2023	68.8	72.8
FocalFormer3D-TTA [11]	ICCV 2023	70.5	73.9
LiDARFormer		68.9	72.4
LiDARFormer-TTA		**71.5**	**74.3**

Table 2: Segmentation results on the `test` split of nuScenes.

Model	Ref	mIoU
PolarNet [74]	CVPR 2020	69.8
PolarStream [6]	NeurIPS 2021	73.4
JS3C-Net [62]	AAAI 2021	73.6
Cylinder3D [81]	CVPR 2021	77.2
AMVNet [33]	arXiv 2020	77.3
SPVNAS [48]	ECCV 2020	77.4
Cylinder3D++ [81]	CVPR 2021	77.9
AF2S3Net [13]	CVPR 2021	78.3
GASN [67]	ECCV 2022	80.4
SPVCNN++ [48]	ECCV 2020	81.1
LidarMultiNet [66]	AAAI 2023	81.4
LiDARFormer		81.0
LiDARFormer-TTA		**81.5**

Table 3: Results on the `val` split of nuScenes. *: Reported by [81].

Model	mIoU	mAP	NDS
RangeNet++ [38]	65.5*	-	-
PolarNet [74]	71.0*	-	-
SalsaNext [15]	72.2*	-	-
AMVNet [33]	77.2	-	-
Cylinder3D [81]	76.1	-	-
RPVNet [60]	77.6	-	-
SphereFormer [26]	78.4		
CBGS [79]	-	51.4	62.6
CenterPoint [69]	-	57.4	65.2
TransFusion-L [1]	-	60.0	66.8
BEVFusion-L [34]	-	64.7	69.3
LidarMultiNet [66]	82.0	63.8	69.5
LiDARFormer seg only	81.7	-	-
LiDARFormer	**82.7**	**66.6**	**70.8**

Table 4: Detection L2 mAPH results on the `test` split of WOD. "L" and "CL" denote LiDAR-only and camera & LiDAR fusion methods. Second best results are underlined.

Model	Ref	Modal	Frame	Veh.	Ped.	Cyc.	Mean
M3DETR [21]	WACV 2022	L	1	70.0	52.0	63.8	61.9
PV-RCNN++ [45]	arXiv 2022	L	1	73.5	69.0	68.2	70.2
CenterPoint++ [69]	CVPR 2021	L	3	75.1	72.4	71.0	72.8
SST_3f [17]	CVPR 2022	L	3	72.7	73.5	72.2	72.8
AFDetV2 [23]	AAAI 2022	L	2	73.9	72.4	73.0	73.1
DeepFusion [31]	CVPR 2022	CL	5	75.7	76.4	74.5	75.5
MPPNet [7]	ECCV 2022	L	16	76.9	75.9	74.2	75.7
CenterFormer [78]	ECCV 2022	L	16	**78.3**	**77.4**	73.2	_76.3_
BEVFusion [34]	ICRA 2023	CL	3	_77.5_	76.4	**75.1**	_76.3_
LiDARFormer		L	3	_77.5_	77.2	_74.6_	**76.4**

Table 5: Results on `val` split of WOD. *: From our reproduction.

Model	Ref	Frame	mIoU	L2 mAPH
PolarNet [74]	CVPR 2020	1	61.6*	-
Cylinder3D [81]	CVPR 2021	1	66.6*	-
SphereFormer [26]	CVPR 2023	-	69.9	-
PV-RCNN++ [45]	IJCV 2022	1	-	68.6
AFDetV2-Lite [23]	AAAI 2022	1	-	68.8
CenterPoint++ [69]	CVPR 2021	3	-	71.6
FlatFormer [35]	CVPR 2023	3	-	72.0
SST [17]	CVPR 2022	3	-	72.4
DSVT [52]	CVPR 2023	3	-	75.5
CenterFormer [78]	ECCV 2022	8	-	73.7
MPPNet [7]	ECCV 2022	16	-	74.9
LidarMultiNet [66]	AAAI 2023	3	71.9	75.2
LiDARFormer seg only	-	3	71.3	-
LiDARFormer	-	3	**72.2**	**76.2**

while full LiDARFormer further improves the mIoU to 82.7% with the SOTA detection performance NDS 70.8%. Our method surpasses all previous state-of-the-art methods, which matches our result in the test set.

Waymo Open Dataset Table 4 shows the detection results of LiDARFormer on the test set of WOD. LiDAR-Former achieves the state-of-the-art performance of 76.4% L2 mAPH, outperforming even the camera-LiDAR fusion methods and methods that use a much greater number of frames. Lastly, we report the validation results on Waymo Open Dataset in Table 5. We reproduce the result of Polar-Net and Cylinder3D based on their released code for comparison. Our segmentation-only LiDARFormer achieves a 71.3% mIoU performance on the validation set. Our multi-task model also outperforms the previous best multi-task network by 0.3% on the segmentation task. For the more competitive detection task, our method reaches the best L2 mAPH result of 76.2%.

4.4. Ablation Study

Effect of Transformer Structure on Segmentation Task Table 6 shows the effectiveness of each proposed component in our method when trained only for the segmentation task. We use the network described in 3.1 as our baseline model. This simple design already can achieve competitive performance compared to other current state-of-the-art methods. After adding the segmentation transformer decoder, the mIoU increases by 1.7% and 0.3% in nuScenes and WOD, respectively. By concatenating points from previous frames to the current frame, the result further increases by 2.5% and 0.6%. The cross-space transformer also can improve the mIoU by 0.9% and 0.1%, respectively.

Effect of the Unified Multi-task Transformer Decoder Table 7 demonstrates the improvements achieved by our proposed transformer decoder in the multi-task network. We use the 1st-stage results of LidarMultiNet [66] as our baseline. Adding an individual transformer decoder to either the detection or segmentation branch results in improved performance in both tasks, as our multi-task network has a shared backbone, allowing improvement in one task to contribute to feature representation learning. Our proposed shared transformer decoder yields superior overall performance by introducing cross-task attention learning. The cross-space transformer module further improves per-

Table 6: The ablation of mIoU improvement of each component on the nuScenes and WOD `val` split when trained only for the segmentation task. XSF and STD stand for cross-space transformer and segmentation transformer decoder.

Baseline (3.1)	STD	Multi-frame	XSF	nuScenes	WOD
✓				76.6	70.3
✓	✓			78.3 (+1.7)	70.6 (+0.3)
✓	✓	✓		80.8 (+4.2)	71.2 (+0.9)
✓	✓	✓	✓	**81.7** (+5.1)	**71.3** (+1.0)

Table 7: The ablation of the improvement of shared transformer decoder on the nuScenes `val` split when jointly trained with detection task.

Baseline [66]	XTF		XSF	mIoU	mAP	NDS
	Seg	Det				
✓				81.8	65.2	70.0
✓	✓			82.1 (+0.3)	65.4 (+0.2)	70.2 (+0.2)
✓		✓		82.4 (+0.6)	65.9 (+0.7)	70.3 (+0.3)
✓	✓	✓		82.6 (+0.8)	66.0 (+0.8)	70.2 (+0.2)
✓	✓	✓	✓	**82.7** (+0.9)	**66.6** (+1.4)	**70.8** (+0.8)

Table 8: Panoptic segmentation result on nuScenes `val` split.

	stage	PQ	SQ	RQ	mIoU
LidarMultiNet [66]	2-stage	**81.8**	**90.8**	89.7	83.6
LiDARFormer	1-stage	**81.8**	90.7	**89.9**	**84.1**

Table 9: Design choice of the segmentation decoder on the nuScenes `val` split.

LiDARFormer seg only result without XSF (mIoU)	80.8
w/o voxel to class attention	80.4 (-0.4)
w/o class to voxel attention	80.1 (-0.7)
w/o dynamic kernel	80.3 (-0.5)
w/o class embedding initialization	80.5 (-0.3)

Table 10: The ablation of XSF on the nuScenes `val` split. S→D and D→S denote sparse-to-dense (3b) and dense-to-sparse (3a) XSFs.

S→D	D→S	Add Convs	mIoU	mAP	NDS
Segmentation Only					
✓	✓		81.7	-	-
		✓	80.9 (-0.8)	-	-
Multi-task					
✓	✓		82.7	66.6	70.8
	✓	✓	82.8 (+0.1)	66.0 (-0.6)	70.5 (-0.3)

Direct Mapping   Cross-space Transformer

Figure 5: **Visualization of the learned offsets.** We showcase the features of a car's 3D voxels (blue) and their corresponding deformable offsets (red) that were learned in our XSF module. For a better visual representation, we only highlight the offsets with high attention scores.

formance, particularly for the detection task. We also evaluate the panoptic segmentation performance of our multi-task network in Table 8. Even without a second stage dedicated to panoptic segmentation, our model achieves competitive results compared to the previous best method, LidarMultiNet. This demonstrates the ability of our multi-task transformer decoder to generate more compatible results for both tasks.

4.5. Analysis

Analysis of the Segmentation Decoder We compare the segmentation-only performance of our method using different transformer designs in Table 9. Removing either way of the cross-attention leads to an inferior result. The dynamic kernel design outperforms the traditional segmentation head by 0.8%. Furthermore, the performance is 0.3% lower without using an auxiliary segmentation head to initialize the class embedding.

Analysis of Cross-space Transformer Table 10 illustrates the effectiveness of the Cross-Space Transformer (XSF) module in both detection and segmentation tasks, as compared to the direct mapping method. If we replace XSF with additional convolution layers of similar parameter size, the segmentation performance decreases by 0.8%. However, when we only replace the sparse-to-dense XSF in the multi-task model, the segmentation performance remains largely unaffected, while detection performance shows a significant decline. This finding suggests that the dense-to-sparse and sparse-to-dense XSFs contribute differently to the detection and segmentation tasks.

In Figure 5, we provide a visualization of the deformable offsets in our cross-space transformer. When using the previous direct mapping method for sparse voxels, only the features in the same position are used for transferring features between 3D and 2D space. This method may not utilize some useful features learned in the dense 2D BEV map. In contrast, our method is capable of aggregating related features across a wider range.

5. Conclusion

In this paper, we present **LiDARFormer**, a novel and effective paradigm for multi-task LiDAR perception. Our

method offers a novel way of strengthening voxel feature representation and enables joint learning of detection and segmentation tasks in a more elegant and effective manner. Although we have designed LiDARFormer for LiDAR-only input, our transformer XSF and XTF can extend to learn multi-modality and temporal features simply through cross-attention layers. Similarly, XSF can apply multi-scale feature maps in the deformable attention module to further extract the contextual information with larger receptive fields. LiDARFormer sets a new state-of-the-art performance on the competitive nuScenes and Waymo detection and segmentation benchmarks. We believe that our work will inspire more innovative future research in this field.

A. Network Details

Voxel Feature Encoder We adopt the same design as [74] to encode the point cloud into a voxel feature map. First, we group points within each voxel together and append 6 additional features to the point features $P \in \mathbb{R}^{3+c+6}$, i.e. the center of corresponding voxel (x_v, y_v, z_v) and the offset to the center $(x - x_v, y - y_v, z - z_v)$. Next, we use 4 stacked layers of MLP to transform the point feature to a high dimensional space, followed by a sparse max pooling layer to extract voxel feature representation in each valid voxel. The channel size is $[64, 128, 256, 256]$ in each MLP.

XSF structure We apply 2 stacked transformer blocks, each with 4 heads of deformable self-attention. We use a channel size of 64 in each head of Dense-to-Sparse XSF and a channel size of 32 in each head of Sparse-to-Dense XSF. The channel size of the FFN is 256 in both XSFs. We use pre-norm rather than post-norm in each layer.

XTF structure We use 3 stacked transformer decoder layers, each with 4 heads of self-attention and cross-attention. We use a channel size of 32 in each head and channel size of 64 in the FFN.

B. More Discussions

More Analysis of XSF In Table 11, we show the results of adopting different types of query in the dense-to-sparse XSF. The features in the BEV feature map at valid voxels serve as queries in our LiDARFormer. Voxel query refers to taking the features from a sparse feature map at valid voxels while embedding query means treating the embeddings of the valid coordinates as queries. The performance slightly drops by 0.3% and 0.4% respectively, which may be due to the BEV features from the 2D multi-scale feature extractor containing more contextual information.

Runtime and Model Size We evaluated the runtime and model size of LiDARFormer on Nvidia A100 GPU. Figure 6 demonstrates that a multi-task network can significantly reduce latency by sharing backbone networks. Our

Figure 6: **Inference Latency Comparison.** The notation "Det + Seg" refers to the combination of latencies from the previous SOTA detection and segmentation methods. Specifically, we chose CenterFormer [78] and Cylinder3D [81] because they share similar backbone network structures with our approach. All methods were evaluated on Nvidia A100 GPU.

Table 11: The ablation of different query types adopted in the dense-to-sparse XSF on the nuScenes `val` split.

LiDARFormer seg only result (mIoU)	81.7
Voxel Query	81.4 (-0.3)
Embedding Query	81.3 (-0.4)

Table 12: Comparison of different class feature embedding initialization methods.

Initialization Method	mAP	NDS	mIoU
BEV	66.6	70.8	82.7
Voxel	66.4	70.5	82.9

approach has similar latency to the previous 2-stage multi-task network but outperforms it in an end-to-end 1-stage network design. Additionally, LiDARFormer employs fewer parameters (77M) than the LidarMultiNet (131M).

Class Feature Embedding Initialization In our cross-task transformer decoder, we initialize the class feature embedding using a coarse prediction and its BEV features. As shown in Table 12, if we change the initialization to use voxel features, the performance of LiDARFormer will increase in the segmentation task but will decrease in the detection task.

Analysis of Cross-task Transformer We compare the segmentation prediction of our model to the baseline model without our proposed transformer module. As shown in Figure 7, we notice that the improvement usually takes place in the discontinuous area, e.g. points from one object are predicted to have labels of different classes.

Polar Coordinate PolarNet [74] and Cylinder3D[81] have shown the potential of polar feature representation on the LiDAR segmentation problem. It mimics the scan pattern of the LiDAR sensor to balance the point distribution across different ranges of voxels. Contrary to their finding, the experiment on Waymo Open Dataset shows the perfor-

Ground Truth   **LidarMultiNet**   **LiDARFormer**

Legend:
- Car
- Barrier
- Driveable
- Vegetation
- Sidewalk
- Terrain
- Manmade
- Pedestrian
- Truck
- Other_flat

Figure 7: **Visualization of the improvement of our model on the nuScenes `val` split.** With our proposed cross-space and cross-task transformer module, our approach generates more accurate labels in the previously uncertain area. Best viewed in color.

Table 13: Segmentation results of each class on the `test` split of nuScenes. We underline the best performance in each category.

Model	mIoU	barrier	bicycle	bus	car	construction vehicle	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surface	other flat	sidewalk	terrain	manmade	vegetation
PolarNet [74]	69.8	80.1	19.9	78.6	84.1	53.2	47.9	70.5	66.9	70.0	56.7	96.7	68.7	77.7	72.0	88.5	85.4
PolarStream [6]	73.4	71.4	27.8	78.1	82.0	61.3	77.8	75.1	72.4	79.6	63.7	96.0	66.5	76.9	73.0	88.5	84.8
JS3C-Net [62]	73.6	80.1	26.2	87.8	84.5	55.2	72.6	71.3	66.3	76.8	71.2	96.8	64.5	76.9	74.1	87.5	86.1
Cylinder3D [81]	77.2	82.8	29.8	84.3	89.4	63.0	79.3	77.2	73.4	84.6	69.1	97.7	70.2	80.3	75.5	90.4	87.6
AMVNet [33]	77.3	80.6	32.0	81.7	88.9	67.1	84.3	76.1	73.5	84.9	67.3	97.5	67.4	79.4	75.5	88.7
SPVNAS [48]	77.4	80.0	30.0	91.9	90.8	64.7	79.0	75.6	70.9	81.0	74.6	97.4	69.2	80.0	76.1	89.3	87.1
Cylinder3D++ [81]	77.9	82.8	33.9	84.3	89.4	69.6	79.4	77.3	73.4	84.6	69.4	97.7	70.2	80.3	75.5	90.4	87.6
AF2S3Net [13]	78.3	78.9	52.2	89.9	84.2	77.4	74.3	77.3	72.0	83.9	73.8	97.1	66.5	77.5	74.0	87.7	86.8
GASN [67]	80.4	85.5	43.2	90.5	92.1	64.7	86.0	83.0	73.3	83.9	75.8	97.0	71.0	81.0	77.7	91.6	90.2
SPVCNN++ [48]	81.1	86.4	43.1	91.9	92.2	75.9	75.7	83.4	77.3	86.8	77.4	97.7	71.2	81.1	77.2	91.7	89.0
LidarMultiNet [66]	81.4	80.4	48.4	94.3	90.0	71.5	87.2	85.2	80.4	86.9	74.8	97.8	67.3	80.7	76.5	92.1	89.6
LiDARFormer	81.0	83.5	39.8	85.7	92.4	70.8	91.0	84.0	80.7	88.6	73.7	97.8	69.0	80.9	76.9	91.9	89.0
LiDARFormer-TTA	81.5	84.4	40.8	84.7	92.6	72.7	91.0	84.9	81.7	88.6	73.8	97.9	69.3	81.4	77.4	92.4	89.6

mance on mIoU has a 1.4% drop when we transform our baseline model to the polar coordinate with a similar voxel size. We conjecture that it is because we already use a relatively small voxel size, which does not induce a huge imbalance of points accumulation in the close-range voxel. Conversely, polar coordinates suffer more distortion in the distant voxels, leading to inferior performance.

Class-wise Segmentation Results on nuScenes In Table 13, we show the class-wise performance of LiDAR-Former on the test set of nuScenes. The best segmentation results of each class are scattered among the top five meth-

ods. This is due to the learning competition among different classes. For example, better performance in "motorcycle" class will cause a drop in "bicycle" class. How to deal with the competition between similar classes is still an unsolved problem.

C. Qualitative Results

We illustrate the qualitative results of LiDARFormer on nuScenes and WOD in Figure 8, 9. Our method can generate accurate semantic predictions in diverse environments.

Figure 8: Visualization of the detection and segmentation results on nuScenes.

Figure 9: Visualization of the detection and segmentation results on Waymo Open Dataset.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, 2022. 2, 3, 7

[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. 6

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 6

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 3

[5] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In *ECCV*, 2020. 7

[6] Qi Chen, Sourabh Vora, and Oscar Beijbom. Polarstream: Streaming lidar object detection and segmentation with polar pillars. In *NeurIPS*, 2021. 7, 10

[7] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *ECCV*, 2022. 7

[8] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. In *NeurIPS*, 2018. 3

[9] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *CVPR*, 2022. 7

[10] Yukang Chen, Jianhui Liu, Xiaojuan Qi, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Scaling up kernels in 3d cnns. In *arXiv*, 2022. 2, 7

[11] Yilun Chen, Zhiding Yu, Yukang Chen, Shiyi Lan, Animashree Anandkumar, Jiaya Jia, and Jose Alvarez. Focalformer3d: Focusing on hard instance for 3d object detection. In *ICCV*, 2023. 7

[12] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 2, 3, 5

[13] 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. 7, 10

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

[15] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving. In *arXiv*, 2020. 2, 7

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3

[17] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *CVPR*, 2022. 7

[18] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *ICCV*, 2021. 2

[19] 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

[20] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. Afdet: Anchor free one stage 3d object detection. In *CVPRW*, 2020. 2

[21] Tianrui Guan, Jun Wang, Shiyi Lan, Rohan Chandra, Zuxuan Wu, Larry Davis, and Dinesh Manocha. M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers. In *WACV*, 2022. 7

[22] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, 2020. 2

[23] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *AAAI*, 2022. 7

[24] Dihe Huang, Ying Chen, Yikang Ding, Jinli Liao, Jianlin Liu, Kai Wu, Qiang Nie, Yong Liu, and Chengjie Wang. Rethinking dimensionality reduction in grid-based 3d object detection. In *arXiv*, 2022. 7

[25] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 6

[26] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *CVPR*, 2023. 7

[27] 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

[28] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *CVPR*, 2023. 2, 3, 5

[29] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, 2016. 3

[30] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018. 2

[31] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *CVPR*, 2022. 7

[32] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 3

[33] Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. In *arXiv*, 2020. 7, 10

[34] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023. 7

[35] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. Flatformer: Flattened window attention for efficient point cloud transformer. In *CVPR*, 2023. 7

[36] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *ICCV*, 2021. 3

[37] Rodrigo Marcuzzi, Lucas Nunes, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Mask-based panoptic lidar segmentation for autonomous driving. In *RA-L*, 2023. 2

[38] Andres Milioto and C Stachniss. RangeNet++: Fast and accurate LiDAR semantic segmentation. In *IROS*, 2019. 2, 7

[39] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *CVPR*, 2021. 3

[40] Duy-Kien Nguyen, Jihong Ju, Olaf Booji, Martin R Oswald, and Cees GM Snoek. Boxer: Box-attention for 2d and 3d transformers. In *CVPR*, 2022. 3

[41] 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. 2

[42] 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. 2

[43] 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. 3

[44] 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

[45] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. In *IJCV*, 2022. 7

[46] 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. 2, 6

[47] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *CVPR*, 2021. 2

[48] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020. 2, 7, 10

[49] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 5

[51] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*, 2021. 6

[52] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, 2023. 7

[53] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 2, 3, 5

[54] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020. 5

[55] Yue Wang and Justin M Solomon. Object dgcnn: 3d object detection using dynamic graphs. In *NeurIPS*, 2021. 7

[56] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, 2018. 2

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

[58] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 2019. 2

[59] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 2, 3

[60] 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, 7

[61] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *CVPR*, 2020. 2

[62] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, 2021. 7, 10

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

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

[65] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *CVPR*, 2021. 3

[66] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. In *AAAI*, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 10

[67] Maosheng Ye, Rui Wan, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Efficient point cloud segmentation with geometry-aware sparse networks. In *ECCV*, 2022. 7, 10

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

[69] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 2, 6, 7

[70] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 3, 5

[71] Liu Ze, Lin Yutong, Cao Yue, Hu Han, Wei Yixuan, Zhang Zheng, Lin Stephen Ching-Feng, and Guo Baining. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 3

[72] 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. 3, 5

[73] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023. 2, 3

[74] 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. 2, 7, 9, 10

[75] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *CVPR*, 2021. 2

[76] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 3

[77] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 2, 4

[78] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *ECCV*, 2022. 2, 3, 6, 7, 9

[79] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. In *arXiv*, 2019. 7

[80] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2, 3, 4

[81] 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. 2, 7, 9, 10

[82] 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. 3