2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

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 need in the LiDAR perception field for 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 achieves state-of-the-art performance on both tasks.**

I. Introduction

LiDAR point cloud detection and semantic segmentation tasks are among the most fundamental tasks in autonomous vehicle perception. With the recent release of the large-scale LiDAR point cloud datasets [2], [3], there has been a surge of interest in integrating these tasks into a single framework. Current methods [4], [1] rely on voxel-based networks with sparse convolution [5], [6] 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 [7], [8], [9] 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 [10], [11], [12], [13], [14] start to exhibit an outstanding performance on 2D image detection and segmentation tasks. Apart from directly replacing the conventional CNN with the transformer encoder [15], [16], various methods [10], [17], [12], [18] 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 [19], [20], [21]. 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

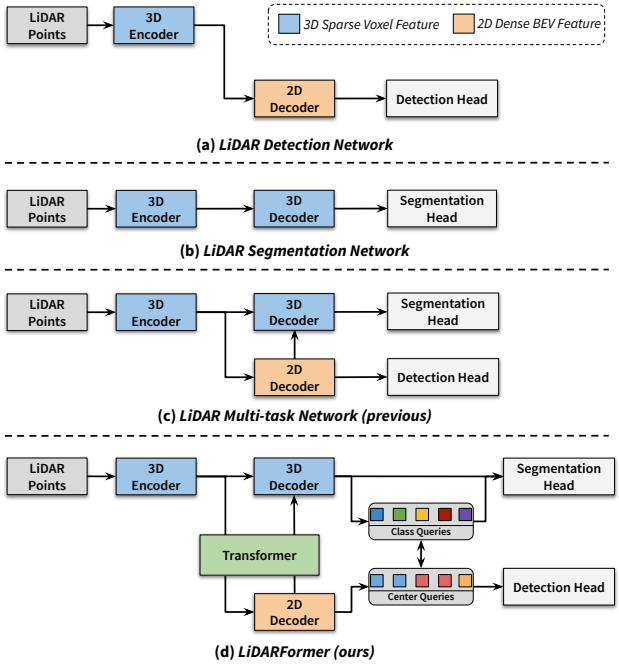


Fig. 1: **LiDAR Perception Network Designs.** LiDAR detection (a) and segmentation (b) networks typically extract feature representations on distinct feature maps. While a recent multi-task network [1] (c) integrates these tasks into a single network, it often overlooks differences among feature maps 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.

[1] Previous work done at TuSimple.
† Contributed equally.

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, which extracts class feature embeddings and refines 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 the challenging nuScenes dataset [2] and the Waymo Open Dataset [3]. 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. LiDARFormer also achieves 76.2% mAPH in the Waymo Open Dataset detection task, 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.

II. Related Work

Voxel-based LiDAR Point Cloud Perception Unlike most point cloud networks [22], [23], [24], [25], [26], [27], [28], [29] 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 [30], [31], 2D BEV [32], [33], [34], or range-view map [35], [36], [37], [38], [39], [40]. Thanks to the development of the 3D sparse convolution layer [5], [6], voxel-based methods are becoming dominant in terms of both high performance and efficient runtime. CenterPoint [41] and AFDet [42] adopted the anchor-free design that detects objects through heatmap classification. LidarMultiNet [1] 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. Some recent methods [8], [7], [43], [9] propose to fuse features from multi-view feature maps, combining point-level information with 2D BEV/range-view and 3D voxel features. 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.

Transformer Decoder Transformer [44] structure has gained huge popularity in recent years. Built on the development of 2D transformer backbones [15], [16], various methods [17], [45], [11], [12], [13], [14] 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 [16], [45], [13] and decoder [10], [11], [12], [46], [14], [18]. 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 [47], [48], [49], [50], [20], [51], [52], [19] 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 [47] and multi-frame [48], [19] / modality [20], [52] feature fusion. However, studying effective methods 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.

III. Method

In this section, we present the design of LiDARFormer. As shown in Figure 2, our framework consists of three parts: (III-A) A 3D encoder-decoder backbone using 3D sparse convolution; (III-B) A Cross-space Transformer (XSF) module extracting large-scale and context features in the BEV; (III-C) A Cross-task Transformer (XTF) decoder that aggregates class-wise and object-wise contextual information from voxel and BEV feature maps. The features are further associated through shared cross-task attention.

A. 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.

As shown in Figure 2, we use the standard VoxelNet [30] as the backbone, where the input point cloud is initially transformed into the voxel space and subsequently processed by 3D sparse convolution layers. 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

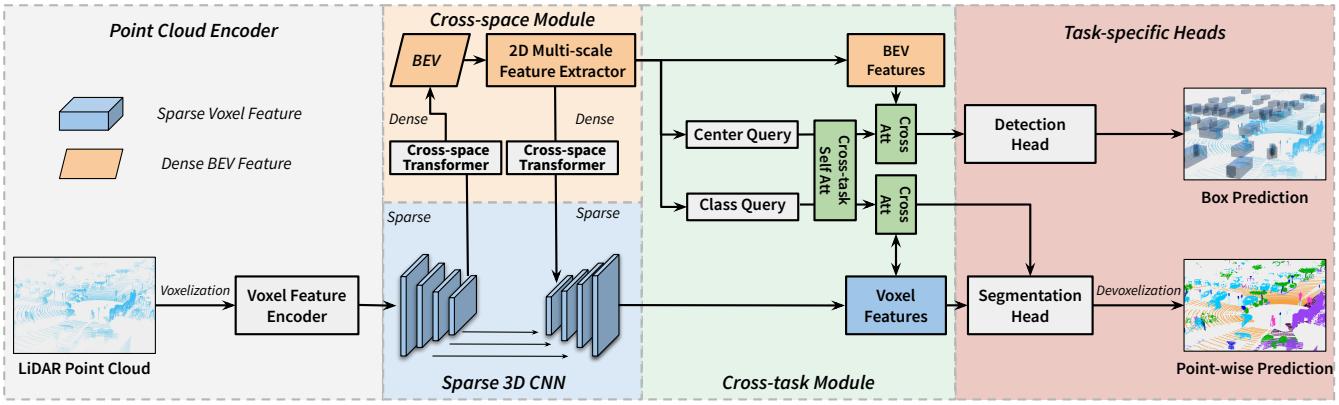
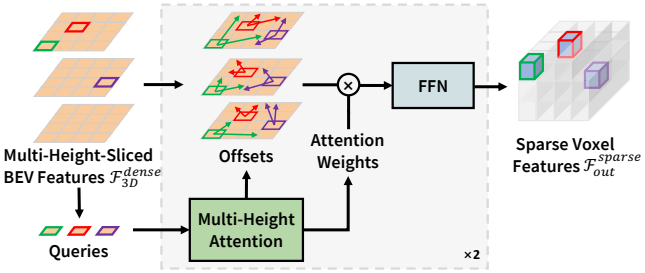
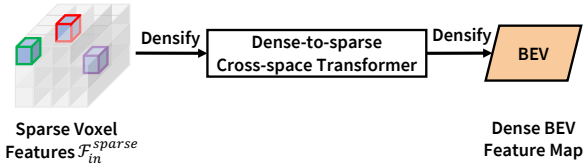


Fig. 2: **The architecture of LiDARFormer.** Our network first transforms the point cloud into a sparse voxel map. Next, sparse 3D CNN is used to extract voxel feature representation. Between the encoder and the decoder, we use a Cross-space Transformer (**XSF**) module to learn long-range information 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.



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



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

Fig. 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.

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.

B. 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 [1] 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.

We employ a cross-space transformer to 1) convert the sparse voxel features in the last scale $\mathcal{F}_{in}^{sparse}$ 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 $\mathcal{F}_{out}^{sparse}$, where d is the downsampling ratio (*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 [17] is adopted as a self-attention layer to explore global information in the dense feature map. Since $\mathcal{F}^{dense}$ lacks height 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.

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.

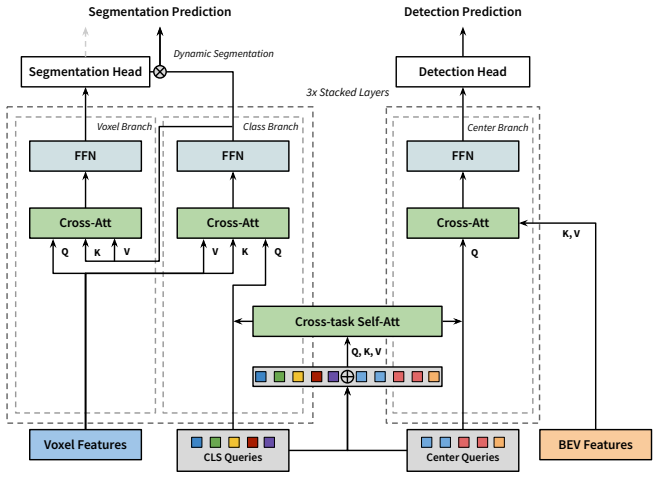


Fig. 4: **Cross-task Transformer (XTF).** The segmentation and detection decoders share a self-attention layer to transfer the cross-task features. In the segmentation decoder, we use a bidirectional cross-attention to refine voxel features based on the aggregated class feature embedding. For simplicity,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

C. Cross-task Transformer Decoder

Although detection and segmentation share correlated information, they are usually learned in two separate network structures. LidarMultiNet [1] demonstrates that through sharing intermediate feature representation, both detection and segmentation performance can be 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 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 $\varepsilon = \{\varepsilon_k | k \in \{1 \dots K\}\}$ 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 [46], we use a transformer decoder to further extract the class feature embedding and refine the 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 [44], containing a 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 $\mathcal{V}$. 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. 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 [53], [11], [12], [18], 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 [19], which represents the object-level feature as center query embedding initialized from BEV center proposals. We initialize the class feature embedding 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.

IV. Experiments

In this section, we present the experimental results of our proposed method on nuScenes dataset [2] and the Waymo Open Dataset[3], and provide a detailed ablation study of the improvements and in-depth analysis of our model.

A. Datasets

NuScenes dataset contains 1000 scenes of 20s video data captured by a 20Hz Velodyne HDL-32E LiDAR. 16 classes are used for the semantic segmentation evaluation. 10 foreground object ("thing") classes are used for the object detection task. For the 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.

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

Model	Ref	mAP	NDS
CBGS [54]	arXiv 2019	52.8	63.3
CenterPoint [41]	CVPR 2021	58.0	65.5
HotSpotNet [55]	ECCV 2020	59.3	66.0
Object DGCNN [56]	NeurIPS 2021	58.7	66.1
AFDetV2 [57]	AAAI 2022	62.4	68.5
Focals Conv [58]	CVPR 2022	63.8	70.0
TransFusion-L [20]	CVPR 2022	65.5	70.2
LargeKernel3D [59]	CVPR 2023	65.3	70.5
SphereFormer [1]	CVPR 2023	65.5	70.7
LidarMultiNet [1]	AAAI 2023	67.0	71.6
MDRNet-TTA [60]	arXiv 2022	67.2	72.0
LargeKernel3D-TTA [59]	CVPR 2023	68.8	72.8
FocalFormer3D-TTA [61]	ICCV 2023	70.5	73.9
LiDARFormer		68.9	72.4
LiDARFormer-TTA		**71.5**	**74.3**

TABLE II: Segmentation results on the `test` split of nuScenes.

Model	Ref	mIoU
PolarNet [34]	CVPR 2020	69.8
PolarStream [62]	NeurIPS 2021	73.4
JS3C-Net [63]	AAAI 2021	73.6
Cylinder3D [31]	CVPR 2021	77.2
AMVNet [64]	arXiv 2020	77.3
SPVNAS [8]	ECCV 2020	77.4
Cylinder3D++ [31]	CVPR 2021	77.9
AF2S3Net [65]	CVPR 2021	78.3
GASN [66]	ECCV 2022	80.4
SPVCNN++ [8]	ECCV 2020	81.1
LidarMultiNet [1]	AAAI 2023	81.4
LiDARFormer		81.0
LiDARFormer-TTA		**81.5**

TABLE III: Results on the `val` split of nuScenes. *: Reported by [31].

Model	mIoU	mAP	NDS
RangeNet++ [39]	65.5*	-	-
PolarNet [34]	71.0*	-	-
SalsaNext [40]	72.2*	-	-
AMVNet [64]	77.2	-	-
Cylinder3D [31]	76.1	-	-
RPVNet [9]	77.6	-	-
SphereFormer [67]	78.4		
CBGS [54]	-	51.4	62.6
CenterPoint [41]	-	57.4	65.2
TransFusion-L [20]	-	60.0	66.8
BEVFusion-L [68]	-	64.7	69.3
LidarMultiNet [1]	82.0	63.8	69.5
LiDARFormer seg only	81.7	-	-
LiDARFormer	**82.7**	**66.6**	**70.8**

Waymo Open Dataset (WOD) contains around 2000 scenes of 20s video that is collected at 10Hz by a 64-line LiDAR. WOD has semantic labels for 23 classes and uses mIoU as the metric. Average Precision Weighted by Heading (APH) is used as the main detection metric. The primary metric mAPH L2 is computed by considering both LEVEL_1 (L1) and LEVEL_2 (L2) difficulty examples.

B. 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 [69]. For the detection task, we followed [41] to use the common center heatmap 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 [70]. 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 [71], [1] were applied when training the model.

C. Main Results

We present the detection and segmentation benchmark results on 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 I and II, we compare LiDARFormer 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 LidarMultiNet, 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 TTA, we also demonstrate the performance on the validation set of nuScenes in Table III. Our segmentation-only LiDARFormer achieves a 81.7% mIoU performance 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 We report the validation results on WOD in Table IV. We reproduce the result of PolarNet and Cylinder3D based on their released code for comparison. Our segmentation-only LiDARFormer achieves a 71.3% mIoU

TABLE IV: Results on `val` split of WOD. *: From our reproduction.

Model	Ref	Frame	mIoU	L2 mAPH
PolarNet [34]	CVPR 2020	1	61.6*	-
Cylinder3D [31]	CVPR 2021	1	66.6*	-
SphereFormer [67]	CVPR 2023	-	69.9	-
PV-RCNN++ [72]	IJCV 2022	1	-	68.6
AFDetV2-Lite [57]	AAAI 2022	1	-	68.8
CenterPoint++ [41]	CVPR 2021	3	-	71.6
FlatFormer [73]	CVPR 2023	3	-	72.0
SST [74]	CVPR 2022	3	-	72.4
DSVT [75]	CVPR 2023	3	-	75.5
CenterFormer [19]	ECCV 2022	8	-	73.7
MPPNet [76]	ECCV 2022	16	-	74.9
LidarMultiNet [1]	AAAI 2023	3	71.9	75.2
LiDARFormer seg only	-	3	71.3	-
LiDARFormer	-	3	**72.2**	**76.2**

TABLE V: 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 (III-A)	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 VI: The ablation of the improvement of shared transformer decoder on the nuScenes `val` split when jointly trained with detection task.

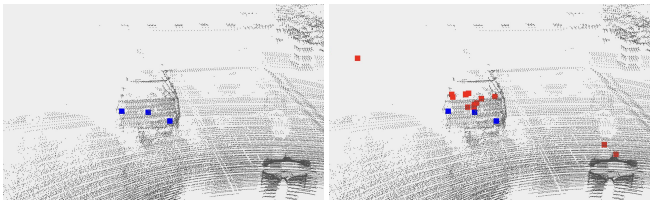
Baseline [1]	XTF Seg	XTF Det	XSF	mIoU	mAP	NDS
✓				81.8	65.2	70.0
✓	✓			82.1 (+0.3)	65.4 (+0.2)	70.2 (+0.2)
✓		✓		82.4 (+0.6)	65.9 (+0.7)	70.3 (+0.3)
✓	✓	✓		82.6 (+0.8)	66.0 (+0.8)	70.2 (+0.2)
✓	✓	✓	✓	**82.7** (+0.9)	**66.6** (+1.4)	**70.8** (+0.8)

TABLE VII: 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 VIII: 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

Fig. 5: **Visualization of the learned offsets.** We showcase a car's 3D voxels (blue) and their deformable offsets (red) learned in the XSF module. To enhance visual clarity, we only highlight the offsets with high attention scores.

performance on the validation set. Our multi-task model also outperforms the previous multi-task network by 0.3% on the segmentation task. For the detection task, our method reaches the L2 mAPH result of 76.2%.

D. Ablation Study and Analysis

Effect of Transformer Structure on Segmentation Task Table V shows the effectiveness of each proposed component in our method when trained only for the segmentation task. We use the network described in III-A 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 VI demonstrates the improvements achieved by our proposed transformer decoder in the multi-task network. We use the 1st-stage results of LidarMultiNet [1] 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 performance, particularly for the detection task.

Analysis of the Segmentation Decoder We compare the segmentation-only performance of our method using different designs in Table VII. Removing either way of the cross-attention leads to an inferior result. The dynamic kernel design outperforms the traditional 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 VIII illustrates the effectiveness of the 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. In the direct mapping method, 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.

V. Conclusion

In this paper, we present a novel and effective paradigm for multi-task LiDAR perception. Our method offers a new 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 contextual information with larger receptive fields. LiDARFormer sets a new state-of-the-art performance standard for public benchmarks. We believe that our work will inspire more innovative future research in this field.

REFERENCES

[1] D. Ye, Z. Zhou, W. Chen, Y. Xie, Y. Wang, P. Wang, and H. Foroosh, "Lidarmultinet: Towards a unified multi-task network for lidar perception," in *AAAI*, 2023.

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

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

[4] D. Feng, Y. Zhou, C. Xu, M. Tomizuka, and W. Zhan, "A simple and efficient multi-task network for 3d object detection and road understanding," in *IROS*, 2021.

[5] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," in *Sensors*, 2018.

[6] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *ICCV*, 2019.

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

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

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

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[11] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Max-deeplab: End-to-end panoptic segmentation with mask transformers," in *CVPR*, 2021.

[12] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *NeurIPS*, 2021.

[13] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *NeurIPS*, 2021.

[14] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," in *ICLR*, 2023.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[16] L. Ze, L. Yutong, C. Yue, H. Han, W. Yixuan, Z. Zheng, L. Stephen Ching-Feng, and G. Baining, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.

[17] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *ICLR*, 2021.

[18] F. Li, H. Zhang, H. xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask dino: Towards a unified transformer-based framework for object detection and segmentation," in *arXiv*, 2022.

[19] Z. Zhou, X. Zhao, Y. Wang, P. Wang, and H. Foroosh, "Centerformer: Center-based transformer for 3d object detection," in *ECCV*, 2022.

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

[21] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, and C. Stachniss, "Mask-based panoptic lidar segmentation for autonomous driving," in *RA-L*, 2023.

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

[23] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, 2017.

[24] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *NeurIPS*, 2018.

[25] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *CVPR*, 2019.

[26] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *CVPR*, 2020.

[27] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann, "Grid-gcn for fast and scalable point cloud learning," in *CVPR*, 2020.

[28] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *ICCV*, 2019.

[29] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *CVPR*, 2021.

[30] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *CVPR*, 2018.

[31] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *CVPR*, 2021.

[32] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *CVPR*, 2018.

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

[34] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *CVPR*, 2020.

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

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

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

[38] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in *ICRA*, 2019.

[39] A. Milioto and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *IROS*, 2019.

[40] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving," in *arXiv*, 2020.

[41] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," in *CVPR*, 2021.

[42] R. Ge, Z. Ding, Y. Hu, Y. Wang, S. Chen, L. Huang, and Y. Li, "Afdet: Anchor free one stage 3d object detection," in *CVPRW*, 2020.

[43] M. Ye, S. Xu, T. Cao, and Q. Chen, "Drinet: A dual-representation iterative learning network for point cloud segmentation," in *ICCV*, 2021.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[45] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *CVPR*, 2021.

[46] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *ECCV*, 2020.

[47] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in *CVPR*, 2021.

[48] Z. Yang, Y. Zhou, Z. Chen, and J. Ngiam, "3d-man: 3d multi-frame attention network for object detection," in *CVPR*, 2021.

[49] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3d object detection via transformers," in *ICCV*, 2021.

[50] H. Sheng, S. Cai, Y. Liu, B. Deng, J. Huang, X.-S. Hua, and M.-J. Zhao, "Improving 3d object detection with channel-wise transformer," in *ICCV*, 2021.

[51] D.-K. Nguyen, J. Ju, O. Booji, M. R. Oswald, and C. G. Snoek, "Boxer: Box-attention for 2d and 3d transformers," in *CVPR*, 2022.

[52] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *ECCV*, 2022.

[53] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic and fast instance segmentation," in *NeurIPS*, 2020.

[54] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," in *arXiv*, 2019.

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

[56] Y. Wang and J. M. Solomon, "Object dgcnn: 3d object detection using dynamic graphs," in *NeurIPS*, 2021.

[57] Y. Hu, Z. Ding, R. Ge, W. Shao, L. Huang, K. Li, and Q. Liu, "Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds," in *AAAI*, 2022.
[58] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3d object detection," in *CVPR*, 2022.
[59] Y. Chen, J. Liu, X. Qi, X. Zhang, J. Sun, and J. Jia, "Scaling up kernels in 3d cnns," in *CVPR*, 2023.
[60] D. Huang, Y. Chen, Y. Ding, J. Liao, J. Liu, K. Wu, Q. Nie, Y. Liu, and C. Wang, "Rethinking dimensionality reduction in grid-based 3d object detection," in *arXiv*, 2022.
[61] Y. Chen, Z. Yu, Y. Chen, S. Lan, A. Anandkumar, J. Jia, and J. Alvarez, "Focalformer3d: Focusing on hard instance for 3d object detection," in *ICCV*, 2023.
[62] Q. Chen, S. Vora, and O. Beijbom, "Polarstream: Streaming lidar object detection and segmentation with polar pillars," in *NeurIPS*, 2021.
[63] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, "Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion," in *AAAI*, 2021.
[64] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, "Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation," in *arXiv*, 2020.
[65] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "(af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," in *CVPR*, 2021.
[66] M. Ye, R. Wan, S. Xu, T. Cao, and Q. Chen, "Efficient point cloud segmentation with geometry-aware sparse networks," in *ECCV*, 2022.
[67] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical transformer for lidar-based 3d recognition," in *CVPR*, 2023.
[68] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *ICRA*, 2023.
[69] M. Berman, A. Rannen Triki, and M. B. Blaschko, "The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *CVPR*, 2018.
[70] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018.
[71] C. Wang, C. Ma, M. Zhu, and X. Yang, "Pointaugmenting: Cross-modal augmentation for 3d object detection," in *CVPR*, 2021.
[72] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection," in *arXiv*, 2022.
[73] Z. Liu, X. Yang, H. Tang, S. Yang, and S. Han, "Flatformer: Flattened window attention for efficient point cloud transformer," in *CVPR*, 2023.
[74] L. Fan, Z. Pang, T. Zhang, Y.-X. Wang, H. Zhao, F. Wang, N. Wang, and Z. Zhang, "Embracing single stride 3d object detector with sparse transformer," in *CVPR*, 2022.
[75] H. Wang, C. Shi, S. Shi, M. Lei, S. Wang, D. He, B. Schiele, and L. Wang, "Dsvt: Dynamic sparse voxel transformer with rotated sets," in *CVPR*, 2023.
[76] X. Chen, S. Shi, B. Zhu, K. C. Cheung, H. Xu, and H. Li, "Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection," in *ECCV*, 2022.