CenterFormer: Center-based Transformer for 3D Object Detection

Zixiang Zhou[1,2], Xiangchen Zhao[1],
Yu Wang[1], Panqu Wang[1], and Hassan Foroosh[2]

[1] TuSimple
[2] Computational Imaging Lab., University of Central Florida
zhouzixiang@knights.ucf.edu
{xiangchen.zhao,yu.wang,panqu.wang}@tusimple.ai
hassan.foroosh@ucf.edu

We present the supplementary material for "CenterFormer: Center-based Transformer for 3D Object Detection" in this paper.

1 Implementation Details

VoxelNet backbone network We adopt the same VoxelNet backbone network design in CenterPoint [7]. In specific, we first use an average pooling in each voxel to encode the point cloud into a voxel feature map. Then, a VoxelNet [8] with sparse convolution is used to extract features in the voxel map. Except for the down-sample layer, all residual blocks use the submanifold sparse convolution layer to minimize the computation cost. The VoxelNet backbone network down-samples the dimensions of the x-axis and y-axis with a factor of 8 and the z-axis with a factor of 16. Finally, the output voxel feature map is reshaped to the BEV for the following process.

Multi-scale CPN We design the Multi-scale CPN to achieve two goals: First, we want to encode the BEV feature into different scale levels for the transformer decoder. Second, the BEV feature map should be large enough to separate small objects like the pedestrian. Since in our experiment, the size of each BEV grid in the VoxelNet output feature is [0.8m $\times$ 0.8m], which is similar to the average pedestrian object size, we need to up-sample the feature map to avoid the voxelization error. We also use a down-sample layer to extract larger-scale features. The overall network structure is shown in Figure 1.

Spatial-aware heatmap fusion To focus the heatmap fusion on the current object center location, we use the current BEV feature as the reference to learn spatial attention. We concatenate the current feature and the weighted previous features together and use another 3×3 convolution layer to compress the BEV feature into the same size as the single frame input in the heatmap head.

Training details Generally, the transformer decoder requires a matching process, e.g. Hungarian matching, to find the closest ground truth bounding box to the prediction in training. The computation cost of this process becomes unacceptable when we match two 3D bounding boxes with orientation. Hence, we use the same training strategy in the center-based object detection network, i.e. only training the network when the proposed center is at the same position

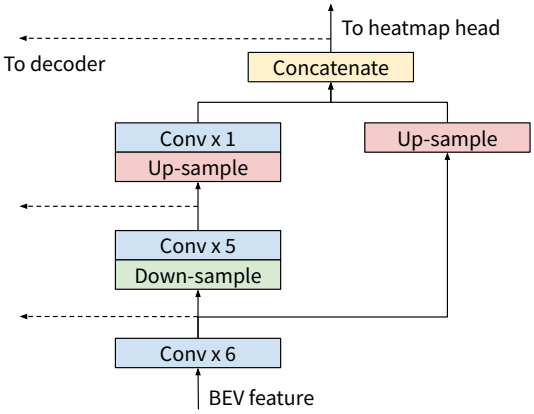


Fig. 1: **The network structure of multi-scale CPN.** Each **Conv** block contains a convolution layer with kernel size 3×3, a batchnorm layer and a relu activation layer. We use convolution layer with stride and transpose convolution layer as the down-sample and up-sample layers.

as the ground truth center. To utilize all annotation information in training, we manually select the center positions of all ground truth bounding boxes as the initial center proposals in training. And the position with the highest heatmap response other than those positions are selected as the remaining proposals. This allows the network to have a meaningful training objective from the beginning of the training, and thus converges faster.

2 NuScenes Dataset result

NuScenes Dataset (ND) [1] is a large-scale dataset created by Motional. It contains 1000 scenes of 20s each, which are split into 700,150,150 sequences for the training, validation and testing. ND uses a 32 lines LiDAR with the frequency of 20 FPS. In each keyframe that is sampled every 0.5s, ND provides bounding box annotations for 10 different classes. The evaluation metrics used by ND are mean average precision (mAP) and nuScenes detection score (NDS). In contrast to WOD, the mAP used by NP only matches objects according to the 2D center distance in BEV rather than IoU. NDS is computed based on a weighted sum of **A**verage **T**ranslation/ **S**cale/ **O**rientation/ **V**elocity and **A**ttribute **E**rrors (ATE/ ASE/ AOE/ AVE/ AAE) on the set of true positives. Followed by [7], we set the range of the 3D voxel space as $[-54\text{m}, 54\text{m}]$ for the X and Y axes, and $[-5\text{m}, 3\text{m}]$ for the Z axis. The size of each voxel is set to $(0.075\text{m}, 0.075\text{m}, 0.2\text{m})$.

We show the comparison of the results on the nuScenes validation set in Table 1. We compare our base CenterFormer model with the CenterPoint baseline using the same training configuration. Due to the time limitation, we did not include further experiments on ND using some of our more complex structures, like deformable cross-attention and multi-frame fusion. Nevertheless, our

Table 1: The detection result on the ND validation set. ‡: Base CenterFormer model without iou rectification and multi-frame fusion. $*$: Our implementation uses the same backbone network and training configuration.

Method	mAP ↑	NDS ↑	ATE ↓	ASE ↓	AOE ↓	AVE ↓	AAE ↓
PointPillars [2]	39.3	53.3	-	-	-	-	-
Pillar-OD [6]	44.4	56.8	-	-	-	-	-
SSN [10]	45.3	57.0	-	-	-	-	-
CBGS [9]	51.4	62.6	-	-	-	-	-
CenterPoint* [7]	55.2	64.4	29.3	25.7	29.6	27.2	**19.5**
CenterFormer‡	**55.4**	**65.2**	**27.5**	**25.2**	**27.5**	**24.3**	20.8

base CenterFormer already outperforms CenterPoint as shown in the table. The improvement comes mainly from the bounding box regression since these two methods share a similar center-based classification design. This result validates the superiority of our proposed CenterFormer over the traditional center-based detector in different point cloud structures.

3 Analysis

The effect of our multi-frame design In Table 2, we show the improvement of our multi-frame CenterFormer compared to the point concatenation method used by most LiDAR object detectors. The point concatenation method has significant improvement (+2.6%) on two frames. But it has less effect when using more frames. In contrast, our multi-frame CenterFormer has constant improvement when using more frames. In 2, 4 and 8 frames, multi-frame CenterFormer achieves 1.0%, 0.5% and 1.1% higher mAPH than the point concatenation method. Our deformable CenterFormer achieves better performance than the base model on multi-frame due to the ability to model cross-attention in a larger range. We also compared the performance on different speeds in Figure 2. The speed of a object is categorized into stationary ($< 0.2m/s$), slow ($0.2 \sim 1m/s$), medium ($1 \sim 3m/s$), fast ($3 \sim 10m/s$) or very fast ($> 10m/s$). We can see that the main improvement in the point concatenation method comes from the stationary objects, and the slow-speed objects even have worse performance. On the contrary, our multi-frame CenterFormer achieves better performance throughout all categories.

Transformer layer and head number We show the comparison of results using different transformer layers and head numbers in Table 3. The results indicate that more transformer layers and heads do not assure better performance. The base transformer model with 3 layers and 4 heads and the deformable transformer model with 2 layers and 6 heads has the best performance.

Cross-attention field We show the comparison of results using different cross-attention window sizes and offset numbers in Table 4. Increasing the window size does not have any performance gain. This is because the sizes of each grid in three scales are $[0.4m, 0.8m, 1.6m]$ in our setting. The 3×3 attention

Table 2: The LEVEL_2 mAPH results comparison of the multi-frame CenterFormer on WOD validation set. All models are trained without the IoU rectification. ⋆: CenterFormer using point concatenation. †: Deformable CenterFormer.

Method	Frame	Vehicle	Pedestrian	Cyclist	Mean
CenterFormer	1	69.0	66.8	68.0	67.9
CenterFormer⋆	2	70.6	70.2	70.7	70.5
CenterFormer⋆	4	71.7	70.8	71.6	71.4
CenterFormer⋆	8	72.0	71.6	71.6	71.7
CenterFormer	2	70.9	70.4	71.8	71.0
CenterFormer†	2	70.7	71.1	72.6	71.5
CenterFormer†	4	71.9	72.2	71.5	71.9
CenterFormer†	8	73.4	73.4	71.7	72.8

Fig. 2: The LEVEL_2 mAPH results comparison breakdown by speed.

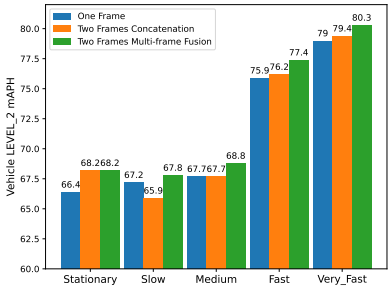


Table 3: The LEVEL_2 mAPH result comparison of using different layer and head configurations on WOD validation set. (Left) Base CenterFormer. (Right) Deformable CenterFormer. **All experiments use only 20% of uniformly sampled training data.**

layer	head	Vehicle	Pedestrian	Cyclist	Mean
2	4	65.5	61.7	63.9	63.7
2	6	65.2	61.0	64.5	63.7
3	2	65.2	61.0	63.3	63.2
3	4	65.4	61.6	65.1	64.0
3	6	65.0	61.6	64.5	63.7
4	2	64.2	60.7	64.1	63.0
4	4	64.9	61.4	64.1	63.5

layer	head	Vehicle	Pedestrian	Cyclist	Mean
1	3	65.1	60.2	64.7	63.3
2	3	64.9	60.7	64.8	63.4
2	6	65.3	60.4	66.0	63.9
3	3	65.7	60.7	65.1	63.7
3	6	64.5	60.3	64.5	63.1

window can encompass the region of almost all pedestrian and cyclist objects. If we increase it to 5×5 or 7×7, although it can include more features for the vehicle, the added feature for the pedestrian and cyclist is almost unrelated. On the other hand, the number of offsets used in the deformable cross-attention also does not increase the performance monotonically. We find the offset number of 15 has the best performance.

Position embedding Position embedding is important in the transformer model to capture the spatial relationship between input features. Standard position embedding is either crafted manually using sine and cosine distances or learned through a linear layer. However, 3D point clouds, as a specific type of data, contain the position information in the raw feature. They do not necessarily need the position embedding since the spatial information is already in the encoded feature. We test the performance of different position embedding methods using 20% of training data. The LEVEL_2 mAPH result is shown in Table 5. We can see without the position embedding, the result drops significantly to 59.5%. This indicates the absolute position information is still an important feature to guide the attention learning of the transformer model. The learnable

Table 4: The LEVEL_2 mAPH result comparison of using different window sizes in our base CenterFormer and different offset numbers in deformable CenterFormer on WOD validation set. (Left) Base CenterFormer. (Right) Deformable CenterFormer. **All experiments use only 20% of uniformly sampled training data.**

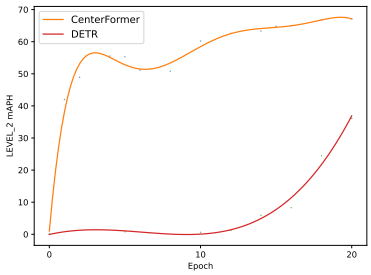
Window size	Vehicle	Pedestrian	Cyclist	Mean
3,3,3	65.4	61.6	65.1	64.0
5,3,3	65.4	61.7	65.0	64.0
5,5,3	65.4	61.8	64.1	63.8
5,5,5	64.1	60.1	63.8	62.7
7,3,3	64.9	61.1	64.7	63.6

Offset number	Vehicle	Pedestrian	Cyclist	Mean
5	64.8	59.5	64.2	62.8
9	64.4	60.6	64.5	63.2
15	65.3	60.4	66.0	63.9
20	65.0	60.2	64.3	63.2

Table 5: The LEVEL_2 mAPH result comparison of the position encoding methods on WOD validation set. **All experiments use only 20% of uniformly sampled training data.**

Encoder	Vehicle	Pedestrian	Cyclist	Mean
None	60.3	56.3	61.3	59.5
Sinusoidal	62.7	58.6	63.5	61.7
Linear	65.2	60.9	66.0	64.0

Fig. 3: The LEVEL_2 mAPH results comparison of CenterFormer and DETR in different epochs.



encoding method also outperforms the sinusoidal encoding method by a large margin.

Converging difficulty compared with DETR In Figure 3, We show the LEVEL_2 mAPH result comparison of CenterFormer and DETR in a 20 epochs training cycle. We implement the DETR-style set matching training strategy using the same backbone network in CenterFormer. We can see that not only CenterFormer can reach a much higher result than DETR, but also converge much faster than DETR.

Comparison with two-stage LiDAR detection Most two-stage LiDAR detection methods [5,4,7,3] apply the RCNN-style refinement network in the 3D domain. The second stage aggregates RoI features in each first-stage proposal to refine both the classification and regression prediction. There are two drawbacks in this design. First, the second stage only utilizes local RoI features. It cannot retrieve global information and depends heavily on the quality of the first feature and proposal. Second, it has a large computation overhead, especially when used in LiDAR point clouds with a large size of points or voxel features. The network will predict the same object information twice, which results in a cumbersome structure and prohibitive run-time. In contrast, our method works between one-stage and two-stage. We use a center proposal network to generate initial center

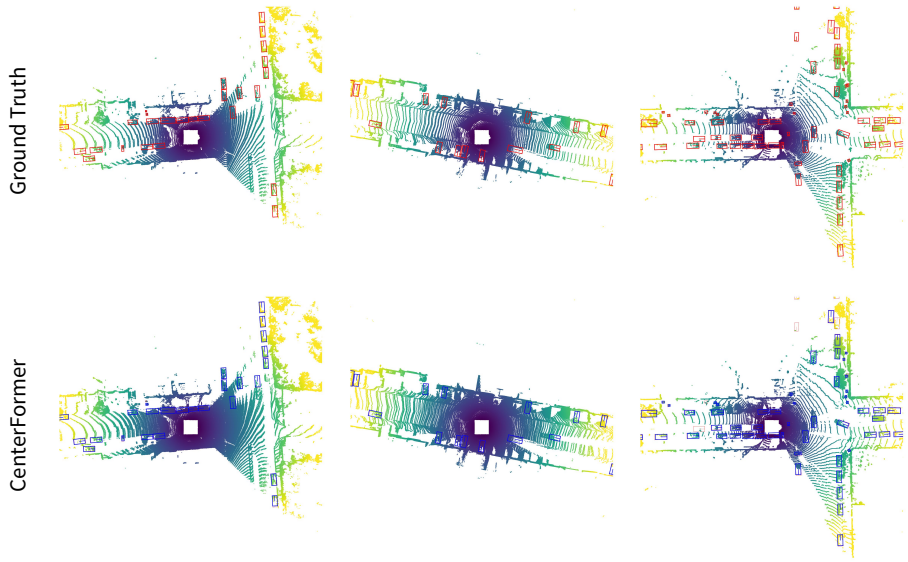


Fig. 4: **Visualization of CenterFormer predictions.** The red box denotes the ground truth bounding box. The blue box denotes the predictions with confidence score > 0.5. Best viewed in color.

queries. The self-attention layer allows the network to directly learn object-level contextual information. The cross-attention layer can also capture long-range information in the multi-scale BEV feature. The classification and regression are done only once in our method.

4 Qualitative Results

Figure 4 shows the qualitative result of our proposed method. Our method can make accurate predictions with a high confidence score.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
2. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: Fast encoders for object detection from point clouds. In: CVPR (2019)
3. Li, Z., Wang, F., Wang, N.: Lidar r-cnn: An efficient and universal 3d object detector. In: CVPR (2021)
4. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR (2020)

5. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object progposal generation and detection from point cloud. In: CVPR (2019)
6. Wang, Y., Fathi, A., Kundu, A., Ross, D.A., Pantofaru, C., Funkhouser, T.A., Solomon, J.M.: Pillar-based object detection for autonomous driving. In: ECCV (2020)
7. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
8. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: CVPR (2018)
9. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. In: arXiv (2019)
10. Zhu, X., Ma, Y., Wang, T., Xu, Y., Shi, J., Lin, D.: Ssn: Shape signature networks for multi-class object detection from point clouds. In: ECCV (2020)