TADP: Task-Aware Deformable Prediction for Single-Stage 3D Object Detection

1st Su Wang
School of Software Engineering
Xi'an Jiaotong University
Xi'an, China
wangsu@stu.xjtu.edu.cn

2nd Yaochen Li*
School of Software Engineering
Xi'an Jiaotong University
Xi'an, China
yaochenli@mail.xjtu.edu.cn

3rd Min Yang
School of Software Engineering
Xi'an Jiaotong University
Xi'an, China
yangmin3056@stu.xjtu.edu.cn

4th Jiaohao Nie
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
jnie002@e.ntu.edu.sg

5th Chang Liu
CSSC Systems Engineering
Research Institute
Beijing, China
liuc1100101110@163.com

6th Yuehu Liu
College of Artificial Intelligence
Xi'an Jiaotong University
Xi'an, China
liuyh@mail.xjtu.edu.cn

***Abstract*—Most single-stage 3D object detectors complete different tasks with the same extracted features. Nevertheless, it is impossible to project features into a common space that is adaptive for all the tasks. We present a novel task-aware deformable prediction (TADP) method for single-stage 3D object detection to solve this problem. Firstly, a triple feature refinement aggregation module is designed to extract three-level features adaptively. Additionally, we design the multi-scale feature aggregation block to fuse multi-scale features in a scale-aware manner. Finally, the prediction of each task is deformed with the designed plug-and-play task-aware deformation head. It can percept the emphasis and interaction of each task. We also designed three different deformation modules. The experimental results demonstrate that the proposed deformation head shows good results on other detection methods. The experimental results on the KITTI dataset demonstrate that the car mAP is 80.91%, surpassing many state-of-the-art methods on the KITTI benchmark.**

***Index Terms*—Point Cloud, Single-stage Detection, 3D Object Detection, Task-aware**

Fig. 1. Visualization of detection in street scenes. It shows the different detection results of SECOND [5] and our TADP. We use red arrows to indicate the biased optimization of our method compared to SECOND detection.

I. Introduction

With the development of automatic driving, lidar sensors are becoming more and more critical in 3D object detection. In general, object detection is divided into two categories in point clouds: single-stage and two-stage. Compared with single-stage methods, two-stage methods have higher accuracy but have more computational costs. Due to the limited power of the embedded hardware for automatic driving cars and robots. We improve the accuracy of the single-stage detector to achieve low computational cost and high accuracy simultaneously. However, there are a large number of sparse and disordered points that pose a significant challenge to the detectors. Some two-stage methods are inspired by PointNet [1] and PointNet++ [2], which through point-based methods to extract scene features. Some single-stage methods are inspired by VoxelNet [3]. These methods encode point clouds orderly, such as voxels and pillars [4].

This work is supported by Key Research and Development Plan of Shaanxi Province (China) under grant no. 2022GY-080.

Previous single-stage methods mainly focus on feature extraction networks instead of the effectiveness of the detection head. The features extracted in the single-stage detector are less accurate than those in the two-stage. Therefore, it is difficult to predict results precisely and may cause some deviation in the difference among tasks in single-stage detectors. Thus, the detection head of tasks should correspond with the task's feature. Therefore, a suitable detection head that can align tasks is crucial for single-stage detectors.

To solve the shortcoming of existing methods, we design a triple feature refinement and aggregation module (TFRA), which refines features in three scales: the semantic, structural, and geometric scales, respectively. In previous methods, directly changing features into different scales and fusing them will cause information loss. Consequently, we design a multi-scale feature aggregation module (MSFA) for feature fusion. Due to the limited accuracy of the single-stage detection head. We propose a task-aware deformable head (TADH). The task perception stack can perceive each task's features and

predict the semantic deformation map (DMap). Moreover, we use height attention to improve the sensitivity of the DMap, which can avoid the limitations of bird-eye view (BEV). Then we modify the prediction results of each task with different suitable deform strategies. Our proposed method dramatically improves the accuracy of the single-stage methods. It is worth mentioning that TADH is detachable and plug-and-play for other 3D detectors, and the experiments show that the accuracy has been improved when TADH is applied to other single-stage detectors. Our primary contribution is manifold.

- We present an efficient and high-precision end-to-end single-stage detection network task-aware called TADP.
- We design three-level feature extraction and scale-aware fusion network to extract and fuse multi-scale features of point cloud scenes effectively.
- We propose a plug-and-play task-aware deformable head. It is adopted to optimize the prediction results of the task. Applying it to other detectors can also significantly improve accuracy.

Our method has achieved high performance and superior inference speed in 3D road scene detection in the KITTI [6] dataset.

II. RELATED WORK

Two kinds of Lidar-based 3D object detectors. (i) The two-stage detectors generate RoI in the first stage and refine RoI in the second stage. (ii) The single-stage object detectors directly generate regression classification and boundary boxes from the first stage. The two-stage method has the advantage of high accuracy. Despite the high accuracy of two-stage, single-stage methods are widely used due to their simpler structure and higher speed. With the development of single-stage methods, the accuracy of the two-stage can be gradually reached. They are becoming an important method for 3D perception in autonomous driving.

PointRCNN [7] is a two-stage detection method based on PointNet++, and the author proposes a point-based Anchor-Free strategy. The first stage proposes regional suggestions, and the second stage refines the interior point features to adjust the 3D frame. PointFormer [8] uses three transformers to extract scene features and refine them. But it has a large computational cost. Voxel R-CNN [9] uses voxel-based rather than point-based methods and adopts voxel RoI pooling to extract proposal features in more detailedly. Votr [10] extracts contextual information between voxels by designing a voxel-based sparse transformer. SST [11] successfully increases the receptive field through the transformer, increasing the accuracy of small objects. However, the two-stage methods have the problems of over-computation and high computational cost.

The single-stage methods include VoxelNet, which first pre-processes disordered point clouds into regular voxels. PointPillar and SECOND segment point clouds into regular voxels and utilize sparse convolution and submanifold convolution to process segmented voxels. TANet [12] designs a triple attention module based on points to extract robust features. 3DSSD [13] devises a more advanced point-based fusion adoption strategy. CIA-SSD [14] proposes a voxel-based IOU-predictive perceptual detection head. However, the features of the single-stage are fragile and the results are unstable.

The proposed task-aware deformation for the detection head can be used to correct the misaligned prediction results. We aim to increase the prediction accuracy of the single-stage method with the controllable computational cost. Enables single-stage detection to be high-speed and high-precision at the same time.

III. METHOD

We design a single-stage object detection method called task-aware deformable prediction for single-stage 3D object detection (TADP). Fig. 2 shows our designed method, which consists of three parts: (i)sparse blocks (SP Blocks); (ii)triple features refine and fusion, which contains tripe feature refinement aggregation (TFRA) and multi-scale feature aggregation (MSFA); (iii)task-aware deformable head (TADH).

A. Point Cloud Feature Encoding

As shown in Fig. 2, the Sparse Blocks (SP Blocks) encode the point clouds. We convert point clouds to voxel format like SECOND. We divide the scene into 40x1600x1400 voxels. The Sp Blocks' details are shown in Fig. 2. To ensure the sparsity of sampled voxels, the Sp Blocks include submanifold sparse convolution [17] and sparse convolution [16] used alternately. We use the bird-eye view (BEV) method to compress the features and the three features after the SP Blocks are used as the input of the next module.

B. Triple Feature Refine Aggregation

This part mainly describes the TFRA module. The three-level structure is used for feature multi-scale refinement, mainly divided into two modules, a three-branch feature refinement module and a Multi-Scale Feature Aggregation module.

1) Triple Feature Refine and fusion:

We are inspired by the feature pyramid. We designed the TFRA module to extract scene information on various scales. We divide the network into three independent branches, respectively extracting semantic, structural and geometric scale features. Details are shown in Fig. 2(b). We refine the features by using the self-correcting layer. SC-Layer is a stack containing two SCConv [18] and a full connected layer, which is able to extract local and global features flexibly and has a variable receptive field. We use deconv to change the feature size, and design the self-residual, upward-residual, and downward-residual connection. The specific process is shown in Fig. 2.

2) Multi-scale feature aggregation: Directly fuse features of different sizes will cause information loss of fragile features. Aiming at this problem, we designed a feature fusion method called Muti-Scale Feature Aggregation (MSFA), as shown in Fig. 2(c). We devised the scale mapping (SM) method to map all other features to one feature. Different scales use different fusion methods to enhance different details. SM function is shown in Eq.1:

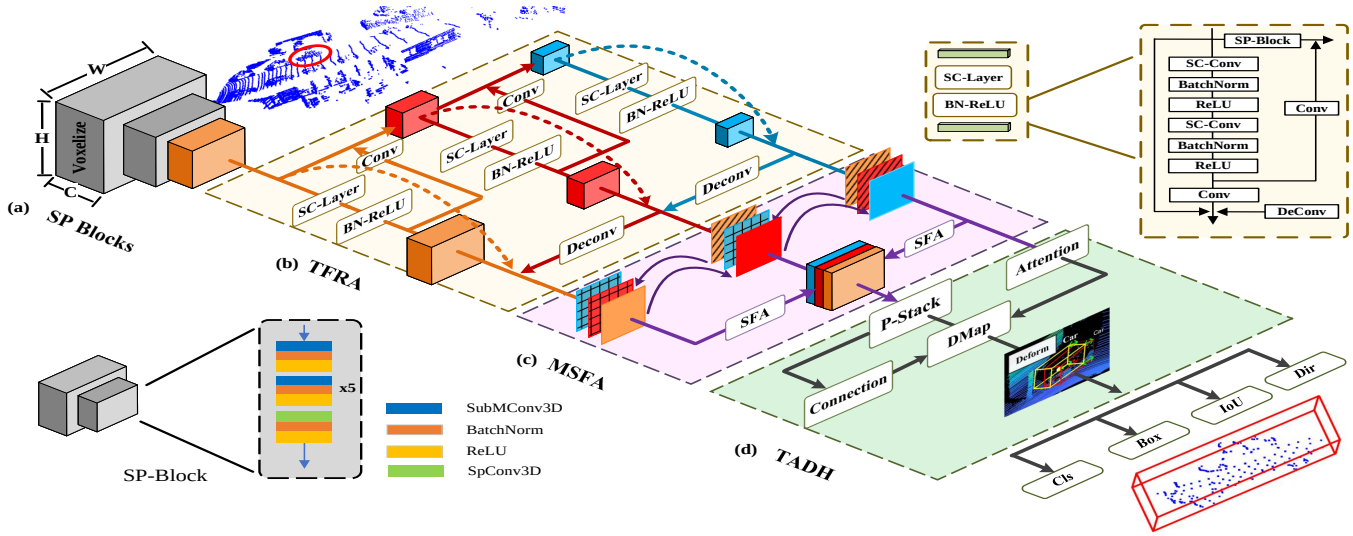

Fig. 2. The figure shows the pipeline of our method TADP. (a) downsample and encode the point clouds. (b) show TFRA structure refine the three-scales features. (c) show the MSFA structure, scale maps Features and fused with SFA. (d) show the TADH, predict the deformation map, and deform the task prediction.

$$M^{x_m,x_n} = P(x_n) \cdot B(C(x_m))$$
$$F_M^{x_m,x_n} = B(C(M^{x_m,x_n} + X_m)) \tag{1}$$

where M^{x_m,x_n} is the feature mapped from x_n to x_m. $C(\cdot)$ stands for Conv. $B(\cdot)$ stands for BatchNorm, $P(\cdot)$ stands for average pooling. $F_S M^{x_m,x_n}$ represents the feature after scale mapping. We scale map each level feature with other features. As shown in Fig. 2, the color represents the feature mapped with, and the pattern represents the meaning of the mapping. SFA uses softmax to establish feature dependencies for adaptive fusion.

C. Task-Aware Defomable Head

The previous detection head can not effectively distinguish the differences between each task, and the predicted features can not precisely match tasks. To solve the problems faced by the single-stage method, we designed a detection head module named Task-Aware Deformable Detection Head (TADH). Firstly, we introduce a task perception stack to sense each task. Then we generate a semantic deformation map (DMap) and introduce extra height attention to sensitize the DMap. Finally, we apply deformation in task prediction. The specific structure is shown in Fig. 3.

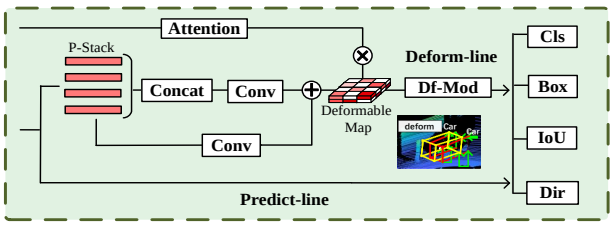

Fig. 3. The pipeline of TADH. The first branch of the features directly generates traditional task predictions through predict line. The second branch through P-Stack generates DMap. The upper features provide height attention. Finally, the deformation to the prediction result is generated through the Deform-line.

1) Task Perceptual Stack: We design a task perceptual stack (P-Stack) to sense task focus and correct misalignment between tasks. The P-Stack contains multiple convolution layers, giving enough mutual receptive fields between tasks. X_0^{perc} represents the refined feature. The P-Stack is composed of N consecutive fully connected layers with activation functions to compute aligned interaction stacks for different features. The perceptual stack formula is shown in Eq.2.

$$X_k^{perc} = B(\sigma(C_k(X_{k-1}^{prec}))) \tag{2}$$

where B, σ represents BatchNorm and ReLU function. C_k represents the k_{th} layer task interactive convolution. X_k^{perc} represents the features stored in the k_{th} perception stack. This stacking feature can well perceive the state of each task by adjusting the dislocation between tasks on the stack. It provides a task-aware pool for the next deformation maps.

2) Deformation Map and Predict Deforme: In order to systematically optimize the prediction results, we design a semantic deformation map (DMap) so that each task learns its own deformation according to the DMap. The method of predicting DMap: First, concat the features of each layer in the P-Stack to generate X^{perc}. Generate DMap using X_N^{perc} residual and X^{perc}. Using the height information of the geometric features to enhance the semantic information of the DMap. The process is shown in Fig. 4. It makes the DMap highly sensitive to compensate for the lack of BEV information.

We design three modules to deform different tasks. The first is the weight module, which generates deformation weights to deform prediction results. The second is the convolution module, which uses deformconv [19] to deform the prediction results. The third is an additional module that adds the deformation to the results. For four tasks: class, bounding box, direction, and IoU, we experiment on different tasks with different deform modules. The results of the experiment are shown in Fig. 5. According to the experimental results, we choose the weight module for the classification task, the

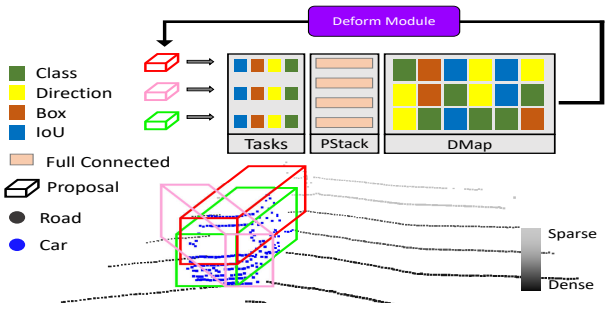

Fig. 4. The process and principle of PStack and DMap generating each task's deformation. Below shows the raw point cloud and proposals. Above shows the process of task alignment

convolution module for the box and direction task, additional module for the IoU task. The algorithm for making prediction deformation is shown in Algorithm 1.

This detection head is plug-and-play, regardless of the height attention attached to the DMap. It is suitable for other single-stage point cloud object detection. TADH can significantly improve the accuracy of other methods.

Algorithm 1 Tasks Deform Module

Note:
$T \in \{box, cls, dir, iou\}$
P_T is the traditional prediction result.
FC_T is task fully connected layer.
DP_T is deformed task prediction.
$DefConv$ is deformconvlution.
procedure TASK PREDICT DEFORM(T, P_T, DM)
  while $t \in T$ **do**
    if $t = cls$ **then**
      $D_{cls} = FC_{cls}(DM)$
      $DP_{cls} = \sigma(\sqrt{P_{cls} * D_{cls}})$
    end if
    if $t = boxort = dir$ **then**
      $D_{box} = FC_{box}(DM)$
      $D_{dir} = FC_{dir}(DM)$
      $DP_{box} = DefConv(P_{box}, D_{box})$
      $DP_{dir} = DefConv(P_{dir}, D_{box})$
    end if
    if $t = iou$ **then**
      $D_{iou} = FC_{iou}(DM)$
      $DP_{iou} = \frac{D_{iou} + P_{iou}}{2}$
    end if
  end while
  return DP_t
end procedure

D. Loss Function

We followed the general setup of loss functions in CIASSD [14] networks. Specifically, we use Focal loss for bounding box classification loss, Smooth-L1 loss for bounding box regression loss, and cross-entropy loss for orientation classification loss. Use SmoothL1 loss for IOU loss. Inspired by gIoU [20], the traditional IoU fails to get correct regression when the anchor and ground truth are at certain angles. We set $\lambda = 1, \mu = 1, \omega = 2, \delta = 0.2$. SML_1 represent Smooth-L1 loss, and total loss L_{total} is as Eq.3:

$$\begin{aligned} L_{giou} &= SML_1(1 - giou) \\ L_{total} &= \lambda L_{cls} + \mu L_{iou} + \omega L_{box} + \delta L_{dri} \end{aligned} \tag{3}$$

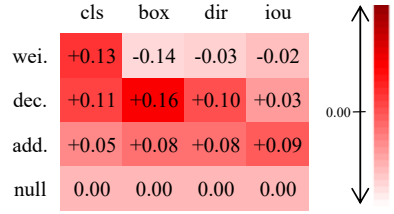

Fig. 5. Comparative experiments between different tasks and different deform modules. *Wei*., *dec*., *add*., and *null*. We show how much the accuracy increases as a color chart. Based on the '*null*' color.

IV. EXPERIMENT

The experiments use the KITTI dataset. We mainly detect car class. The KITTI dataset divides the model evaluation difficulty into Easy, Moderate, and Hard levels. The comparison experiment uses the test set and submits it on the KITTI benchmark. Ablation experiments are evaluated using the validation set because of restricted access to the test set. Fig. 6 shows the visualization of the detection results. According to the official KITTI evaluation metric, the 3D and BEV detection results are evaluated with mean average precision (mAP).

A. Implementation Details

To achieve efficient detection, we adopt data augmentation to eliminate similar classes. We filter out objects in difficulty levels that belong to something other than easy, moderate, and hard to improve the quality of positive samples. Then, taking similar classes of objects (such as van for car) as mitigation, the target of model confusion during training. The selected voxel sizes are [0.05m, 0.05m, 0.1m]. Therefore, the resulting voxel grid size is$1408 \times 1600 \times 40$.

The SC-layer in TFRA selects the $k3$ mode and applies the linear interpolation method for up-sampling. Using a 3×3 convolution kernel and a 5×5 convolution kernel to expand the receptive field. Then, a 1×1 convolution layer is used, and the size of each level is unified with deconvolution for triple fusion. P-Stack in TADH we set N=4. We examine the influence of parameter changes of N on the results through comparative experiments shown in Tab. IV. It can be seen that the accuracy keeps increasing with the increase of N, but the slope gradually decreases. Considering the accuracy and computational cost, we choose $N = 4$. We set the batch size to 4, trained on RTX3090 GPU, and set the epoch to 60. The Adam optimizer is used, the initial learning rate is set to 0.003, and the exponential decay factor is 0.4, which decays every ten cycles.

B. Compared with State-of-The-Art Methods

We compare our algorithm with state-of-the-art 3D road scene detection algorithms. As shown in Tab. I, TADP is compared with the state-of-the-art object detection methods for 3D road scenes in the table. The best effect in the single-stage is shown in bold. It can be seen from the table that our method ranks first in the easy and hard levels of car detection, with 88.93% and 74.17%. Better than all demonstrated single-stage

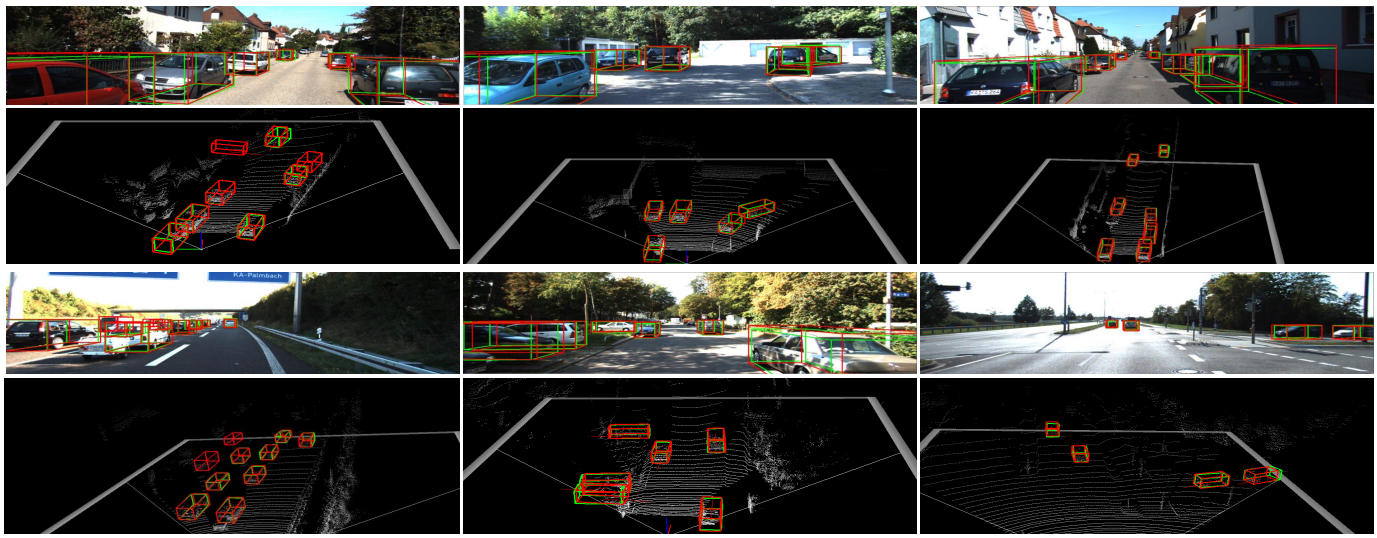

Fig. 6. Our 3D detection results on the KITTI validation set are visualized. The ground truth is the green box, and the prediction box is the red box. The 3D detection box is projected to an RGB image.

TABLE I
AP11-BASED COMPARATION WITH SOTA ON KITTI DATASET BENCHMARK FOR CAR CLASS DETECTION

Type	Method	Sens	$AP_{3D}(\%)$ Easy	Mod	Hard
2-stage	AVOD(2018) [21]	LIDAR+RGB	83.07	71.76	65.73
	PI-RCNN(2020) [22]	LIDAR+RGB	84.37	74.82	70.03
	PointRCNN(2019)	LIDAR	86.96	75.64	70.70
	F-ConvNet(2019) [23]	LIDAR+RGB	87.36	76.39	66.69
	UberATG-MMF(2019)	LIDAR+RGB	88.40	77.43	70.22
	Part-A2(2020) [24]	LIDAR	87.81	78.49	73.51
	Pointformer(2021)	LIDAR	87.13	77.06	69.25
	3D-CVF(2020) [25]	LIDAR+RGB	88.84	79.72	72.80
	Sem-Aug(2022) [31]	LIDAR+RGB	86.69	78.06	73.85
	Fast-CLOCs(2022) [27]	LIDAR+RGB	*89.10*	*80.35*	*76.99*
1-stage	VoxelNet(2018)	LIDAR	79.62	65.97	59.71
	ContFuse(2018) [28]	LIDAR+RGB	83.68	68.78	61.67
	SECOND(2018)	LIDAR	83.34	72.55	65.82
	PointPillars(2019)	LIDAR	82.58	74.31	68.99
	TANet(2020)	LIDAR	84.39	75.94	68.82
	3DSSD(2020)	LIDAR	88.36	79.57	74.05
	SASSD(2020) [29]	LIDAR	88.75	79.79	74.16
	HVPR(2021) [30]	LIDAR	86.38	78.22	73.84
	MGAF(2021) [14]	LIDAR	88.16	79.58	72.39
	ACDet(2022)	LIDAR	88.47	78.85	73.86
	IA-SSD(2022) [15]	LIDAR	88.34	**80.13**	74.04
	TADP(ours)	LIDAR	**88.93**	79.65	**74.17**

TABLE II
ABLATION EXPERIMENTS OF TFRA AND MSFA AND TADH IN TADP ON KITTI VALIDATION DATASET.

TFRA	*MSFA*	*TADH*	*TADH*†	$AP_{3D}(\%)$ Easy	Mod	Hard
				87.52	77.21	74.36
✓				88.27	78.41	75.89
✓	✓			89.26	79.37	76.76
✓	✓	✓		89.71	79.95	77.31
✓	✓		✓	**90.03**	**80.62**	**78.64**

† : means the TADH use GIoU

detectors in easy and moderate levels. TADP outperforms most of the two-stage object detectors in the table above, such as Pointformer and other state-of-the-art detectors. As seen from Tab. III, our method can outperform state-of-the-art two-stage detectors in both running speed and average precision.

TABLE III
RUNTIME (IN MILLISECONDS) AND MAP (IN AP11) COMPARED TO RECENT STATE-OF-THE-ART SECONDARY DETECTORS

	PointRCNN	Part-A2	MGAF-3DSSD	3D-CVF	Ours
time(ms)	643	80	80	85	**40.53**
mAP	77.67	79.7	80.51	80.79	**80.91**

TABLE IV
COMPARITIVE TEST OF CONVOLUTION LAYER PARAMETER N IN PERCETION STACK

Stack_num	0	1	2	3	**4**	5	6
Improvement/$AP_{3D}(\%)$	78.41	+0.11	+0.23	+0.38	**+0.49**	+0.58	+0.64

C. Ablation Experiments

We conduct ablation experiments on the validation set of KITTI. Where *TADH*† in Tab. II represents TADH with GIoU. The data in Tab. II shows that our designed TADH improves the overall network by 0.55%, 0.78%, and 0.76% for easy, moderate, and hard levels, respectively. Moreover, we changed to gIoU for better results, with easy, moderate, and hard levels increasing by 0.32%, 0.67%, and 1.33%, respectively. MSFA improved the easy, moderate and hard classes by 0.99%, 0.96%, and 0.87%, respectively, in the experiment. It can be seen from the table that TADH can greatly optimize the detection results for moderate and hard classes. MSFA can better extract multi-scale features, up to 1% improvement for easy class.

D. TADH Comparison Experiments

In order to verify whether the TADH head can improve the detection effect of the single-stage detector, we spliced TADH into other backbones. We used SECOND, VoxelNet, and TANet networks for comparative experiments on the KITTI validation set. As can be seen from Tab. V, our TADH can significantly improve the detection effect of the backbone network in the one-stage method. This proves that our task deformation head can effectively correct the predicted

TABLE V
COMPARATIVE TEST OF CAR CLASS DETECTION AFTER TADH INSERTION IN SOME ONE-STAGE NETWORKS.

Method	$AP_{3D}(\%)$		
	Easy	Mod	Hard
SECOND	83,52	73.63	67.21
SECOND+DH	**84.44**	**74.25**	**67.61**
Improvement	+0.92	+0.62	+0.40
VoxelNet	79.62	65.97	59.71
VoxelNet+DH	**80.78**	**66.82**	**60.35**
Improvement	+1.16	+0.85	+0.64
TANet	85.42	76.34	69.92
TANet+DH	**86.40**	**77.06**	**70.39**
Improvement	+0.98	+0.72	+0.47

misalignment results and be sensitive to capturing distant features of 3D objects.

V. CONCLUSIONS

This paper proposes a new point cloud single-stage task-aware deformable detector. To solve the situation that most methods of detection tasks are not aligned. Our main contribution includes triple-level extract and aggregate 3D features with multiple scales. It is worth noting that we propose a plug-and-play TADH, which predicts a sensitive deformation map to deform predicted results and reduce the misalignment of features in all tasks. Experiments show that our TADP achieves a really high accuracy performance on the KITTI benchmark. Moreover, the designed plug-and-play TADH can greatly improve the detection accuracy of existing single-stage detectors.

REFERENCES

[1] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77-85.

[2] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems,2017,30.

[3] Y. Zhou and O. Tuzel, VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490-4499.

[4] L. A. H., S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, PointPillars: Fast Encoders for Object Detection From Point Clouds, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12689-12697 .

[5] Y. Yan, Y. Mao and B. Li, SECOND: Sparsely Embedded Convolutional Detection. Sensors (Basel, Switzerland) 18, 2018.

[6] A. Geiger, P. Lenz and R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354-3361.

[7] S. Shi, X. Wang and H. Li, PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019, pp. 770-779.

[8] X. Pan, Z. Xia, S. Song, L. E. Li and G. Huang, 3D Object Detection with Pointformer, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 7459-7468.

[9] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang and H. Li, Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection, arXiv abs/2012.15712, 2021.

[10] J Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu and C. Xu, Voxel Transformer for 3D Object Detection, IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3144-3153.

[11] L. Fan, Z. Pang, T. Zhang, Y. Wang, H. Zhao, F. Wang, N. Wang and Z. Zhang, Embracing Single Stride 3D Object Detector with Sparse Transformer, arXiv abs/2112.06375, 2021.

[12] L. Zhe, X. Zhao, T. Huang, R. Hu, Y. Zhou and X. Bai, TANet: Robust 3D Object Detection from Point Clouds with Triple Attention, AAAI, 2020.

[13] Z. Yang, Y. Sun, S. Liu and J. Jia, 3DSSD: Point-Based 3D Single Stage Object Detector, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11037-11045.

[14] Z. Wu, W. Tang, S. Chen, L. Jiang and C. Fu, CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud, AAAI, 2021.

[15] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan and Y. Guo, Not All Points Are Equal: Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds, arXiv abs/2203.11139, 2022.

[16] B. Liu, M. Wang, H. Foroosh, M. F. Tappen and M. Pensky, Sparse Convolutional Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 806-814.

[17] B. Graham, M. Engelcke and L. V. D. Maaten, 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 9224-9232.

[18] D. Li, C Wang and X Li, Involution: Inverting the Inherence of Convolution for Visual Recognition, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 12316-12325.

[19] X. Zhu, H. Hu, S. Lin and J. Dai, Deformable ConvNets V2: More Deformable, Better Results, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp: 9300-9308.

[20] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 658-666.

[21] J. Ku, M. Mozifian, J. Lee, A. Harakeh and S. L. Waslander, Joint 3D Proposal Generation and Object Detection from View Aggregation, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1-8.

[22] X. Liang, C. Xiang, Z. Yu, G. Xu, Z. Yang, D. Cai and X. He, PI-RCNN: An Efficient Multi-sensor 3D Object Detector with Point-based Attentive Cont-conv Fusion Module, AAAI (2020).

[23] Z. Wang, and K. Jia, Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp: 1742-1749.

[24] S. Shi, Z. Wang, X. Wang and H. Li, Part-A2 Net: 3D Part-Aware and Aggregation Neural Network for Object Detection from Point Cloud, ArXiv abs/1907.03670, 2019.

[25] J. K. Yoo, Y. Kim, J. S. Kim and J. W. Choi, 3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-View Spatial Feature Fusion for 3D Object Detection, ECCV, 2020.

[26] J. Li, H. Dai, L. Shao and Y. Ding, Anchor-free 3D Single Stage Detector with Mask-Guided Attention for Point Cloud, Proceedings of the 29th ACM International Conference on Multimedia, 2021.

[27] S. Pang, D. Morris and H. Radha, Fast-CLOCs: Fast Camera-LiDAR Object Candidates Fusion for 3D Object Detection, IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 3747-3756.

[28] M. Liang, B. Yang, S. Wang and R. Urtasun, Deep Continuous Fusion for Multi-sensor 3D Object Detection, ECCV, 2018.

[29] C. He, H. Zeng, J. Huang, X. -S. Hua and L. Zhang, Structure Aware Single-Stage 3D Object Detection From Point Cloud, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11870-11879.

[30] J. Noh, S. Lee and B. Ham, HVPR: Hybrid Voxel-Point Representation for Single-stage 3D Object Detection, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14600-14609.

[31] L. Zhao, M. Wang, and Y. Yue. "Sem-aug: Improving camera-lidar feature fusion with semantic augmentation for 3d vehicle detection." IEEE Robotics and Automation Letters 7.4 (2022): 9358-9365.