2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Hybrid Point-Pillar-Transformer Network for 3D Small Object Detection in Autonomous Driving

Rongjie Wang[1] and Shuo Yang[2]

[1] College of Computer Science (College of Software), College of Artificial Intelligence,Inner Mongolia University,Inner Mongolia,China

Abstract. With the increasing demand for object detection accuracy in scenarios such as intelligent transportation and autonomous driving, methods that utilize point cloud and pillar features to achieve deep feature fusion have become increasingly common in 3D object detection. However, existing methods often rely on inefficient linear fusion strategies during the process of fusing different features, which fails to adequately capture the dependencies between multi-source features, leading to insufficient feature integration. Additionally, during feature extraction, limitations in network architecture result in a lack of interaction between shallow and deep features, causing the loss of fine-grained feature information, which particularly affects the detection of small objects. To address the above issues, we propose the Hybrid Point-Pillar-Transformer Network (HPPTNet), a two-stage object detection framework that integrates point and pillar features. Specifically, we designed a fine-grained pillar feature extraction module (CFPEM) that effectively alleviates the loss of pillar features caused by downsampling through shallow and deep feature interactions and a lightweight attention design. Next, we developed a transformer-based multi-scale feature fusion module (TMFFM) that dynamically learns the associations between different features through a multi-head attention mechanism, enhancing the ability to capture context-aware features and fully realizing multi-source feature fusion. Experiments on the KITTI dataset demonstrate that our proposed algorithm achieves competitive detection performance compared to several state-of-the-art methods, particularly in the Cyclist and Pedestrian categories. Our code will be open-sourced soon.

Keywords: 3D Object Detection, Feature fusion, Point-Pillar Features.

1 Introduction

3D object detection refers to the precise identification and localization of objects within a three-dimensional scene using point cloud data obtained from LiDAR. As a fundamental task in 3D scene understanding, this technology leverages the depth information derived from point cloud data, enhancing environmental perception capabilities. It has found extensive applications in critical fields such as autonomous driving and robotics.Among the prevailing methodologies, Point-based methods [1] [2] directly extract features from the raw point cloud, capturing fine-grained details while preserving its geometric integrity.Nonetheless, the unstructured, sparse, and irregular nature of point clouds poses significant challenges for effective processing and extension.

On the contrary, grid-based methods transform unstructured point cloud data into structured voxel representations, effectively storing the point cloud in a hash table format.This approach enables the effective utilization of sparse 3D convolutions for efficient feature extraction, significantly enhancing detection efficiency and performance. Specifically, pillar-based methods [3] [4], transform raw point clouds into 2D bird's eye view (BEV) feature maps during voxelization, utilizing a pillar-based pipeline for the 3D object detection task. By projecting the 3D point cloud into a top-down 2D pseudo-image, this method further optimizes computational efficiency .However, the process of quantizing into regular pillars inevitably leads to the loss of fine-grained positional and geometric information, which can impact the accuracy of object detection.

Recently, many 3D detection frameworks adopt a fusion of point and voxel approache [5] [6] for object detection. This hybrid architecture effectively merges voxel methods' efficiency with point methods' precision, demonstrating significant potential in enhancing detection accuracy and robustness.Currently, most methods first extract voxelized features and then combine these voxelized features with keypoints to achieve the fusion of point and voxel features. Compared to single-stage methods, these approaches have significantly improved accuracy. However, they overlook the fact that the sparsity introduced during voxelization and down-sampling convolutions can lead to substantial loss of 3D object features, particularly fine-grained details. This results in extracted voxelized features missing critical information, especially for small objects. Additionally, the fusion of different types of features often involves simple concatenation or weighted summation, without considering the varying importance of different modalities. This coarse-grained fusion approach may fail to fully leverage the correlations and complementarities among different feature types. In practical object detection, this can lead to poor detection performance for pedestrians and cyclists, who occupy only about 0.1% to 0.5% of the entire 3D scene.

Considering the issues outlined above, we propose a novel hybrid detection architecture that combines point and pillar features. In this architecture, to combat the sparsity of point clouds and address potential feature loss during the downsampling process, we innovatively introduce a compact and fine-grained pillar feature extraction module.The module facilitates comprehensive interaction between the extracted shallow and deep pillar features and incorporates a simple attention mechanism to achieve hierarchical feature fusion. This significantly enhances the model's detail capture and overall perceptual capability without substantially increasing computational complexity. Additionally, to fully leverage the extracted multi-class features and learn the dependencies between different modalities for richer information representation, we propose a novel transformer-based multi-scale feature fusion module. This module utilizes different attention heads to capture diverse dependencies among various feature sources and employs a self-attention mechanism to efficiently fuse features based on the importance of each source. We evaluate the proposed architecture on the challenging KITTI dataset, with extensive experiments demonstrating its superior performance over existing methods.

In summary, the main contributions of this study are summarized as follows:

(1) We propose a compact fine-grained pillar feature extraction module that implements novel hierarchical pillar feature interaction. Through shallow-deep feature fusion and a lightweight attention design, it significantly alleviates the feature loss

issues encountered in traditional voxelization methods during the downsampling process.

(2) We introduce a transformer-based multi-head attention mechanism into the pointpillar multi-class features fusion process. By utilizing independent attention heads to capture diverse feature dependencies, such as geometric and semantic relationships, we effectively integrate multi-class features, enhancing the model's ability to understand and capture complex dependencies.

(3) Extensive experiments conducted on the KITTI object detection dataset validate the effectiveness of the HPP-TNet, demonstrating competitive performance in detecting cars, pedestrians, and cyclists.

2 Related work

2.1 Single-Class Feature 3D Object Detection

Single-category 3D object detection has been thoroughly explored in previous research, broadly categorized into point-based methods and voxel-based methods. Point cloud-based methods process the original input point cloud data directly without transformation. For instance, PointNet. [1] was the first to introduce a multi-layer perceptron (MLP) structure that incorporates permutation invariance to directly extract features from point clouds. Building on PointNet, PointNet++ [7] introduced a hierarchical structure that better captures both local and global features of point clouds. PointRCNN [8], leveraging the feature extraction capabilities of PointNet++, proposed an efficient two-stage object detection framework that generates high-quality candidate boxes, refining regression and classification, which effectively improves detection accuracy. While these point-based methods can learn rich spatial features from the original point clouds, they come with high computational costs.

In contrast, voxel-based methods convert irregular and sparse point clouds into regular voxel grids, allowing standard 3D convolution operations to be directly applied to the voxelized data, significantly improving feature extraction efficiency and expressive capability. VoxelNet [9] employs a voxel grid and a deep neural network with 3D convolution for feature extraction but faces high computational costs. SECOND [10] utilizes sparse 3D convolution to avoid redundant calculations on empty voxels, significantly reducing the computational complexity of 3D convolution. PointPillar [3] transforms point cloud data into a different regularized pillar structure, enabling the use of efficient 2D convolution networks, thus providing a lightweight solution for real-time 3D perception systems. Voxel R-CNN [11] introduces a voxel-based two-stage detection framework, achieving refined feature extraction within a voxel-based network.

2.2 Multi-Class Feature 3D Object Detection

The hybrid representation-based works leverage the advantages of both points and voxels for enhanced perception.PV-RCNN [5] integrates multi-scale 3D voxel and BEV features at sampled key points for refinement in the second stage.HVPR [12] enhances point cloud features through a memory module in a hybrid single-stage network, while introducing a multi-scale feature module to effectively address complex scale

variations between objects. HPV-RCNN [13] progressively optimizes candidate boxes by constructing a cascaded subnetwork and designs a partial feature pyramid network to efficiently integrate multi-scale BEV features.APVR [14] proposes an accelerated point-voxel representation method that retains more fine-grained feature information by adding offsets to query neighboring voxels of key points.VFL3D [15] designed a lightweight multi-branch cross-sparse convolution network to improve feature extraction efficiency while maintaining feature granularity. Additionally, it further optimizes BEV features by introducing a compact fine-grained self-attention mechanism, enhancing detection accuracy.

2.3 Transformer-based 3D object detection

The Transformer model was initially used in natural language processing (NLP) and has now demonstrated excellent performance in the fields of computer vision and 3D point cloud processing. Point-Former [16] designs a local-global Transformer architecture that further captures the dependencies between multi-scale representations.Voxel-Former [17] efficiently captures the dependencies between global and local features in point clouds by dividing the point cloud into voxels and integrating the self-attention mechanism of Transformers, enriching the feature representation. PVTransformer [18] aggregates point clouds into voxels and replaces the pooling method of PointNet with an attention mechanism to address the information bottleneck, thereby enhancing the accuracy and scalability of 3D detection.

3 Method

In this paper, we propose a novel two-stage 3D object detection network called the Hybrid Point-Pillar-Transformer Network, which fuses point and pillar features. The overall structure of the Hybrid Point-Pillar-Transformer Network is shown in Fig. 1 . Our model can be roughly divided into several parts: Point Features Extraction, Multi-Scale Pillar Features Extraction, Transformer-based Point-Pillar Feature Fusion Module, and Proposal Generation and Refinement Network. The following subsections provide detailed information about these components.

3.1 Point Features Extraction

The advantage of directly extracting features from the original point cloud is that it effectively retains the original 3D structural information. However, the sheer number of points in the original point cloud means that not all point cloud features are equally important for detection. Retaining more foreground points is essential to capture valuable spatial and positional information. To address this, we consider first sampling key points from the original point cloud P, then using the features of the extracted key points for subsequent detection tasks, achieving GPU acceleration without compromising detection performance. We adopt a sectorized proposal-centric strategy [5]. Specifically, we first use proposal-centric filtering to retain points close to the proposal boxes, effectively reducing the number of candidate points. Next, we apply sectorized sampling to divide the filtered point set into multiple sectors, performing farthest point sampling

(FPS) independently within each sector. Finally, we merge the sampling results from each sector to form the final set of key points.

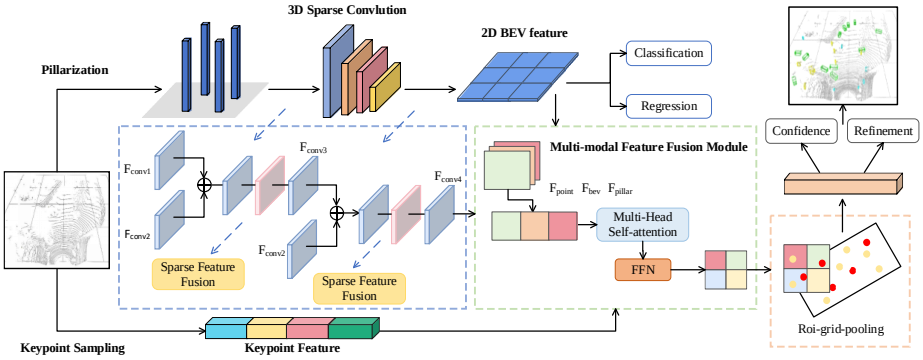

Fig. 1. Overall architecture of the proposed model HPV-TNet.

3.2 Multi-Scale Pillar Features Extraction

After obtaining the features of the point cloud, we construct a pillar structure using a new branch to extract pillarized features. Specifically, given the 3D space defined by range H,W and D along the X-axis, Y-axis, and Z-axis, we divide the space into grids along the X-axis and Y-axis and then stretch each grid along the z-axis to cover the entire z-axis space, resulting in a pillar. We define each pillar to be of size (v_H, v_W, D) .This way, the space is evenly divided into individual pillars, with all points in the sample included in their respective pillars,pillars without any points are considered empty pillars.After obtaining the point cloud tensor represented by pillars, we perform a Max Pooling operation on each non-empty pillar to generate initial features, which are then fed into a 3D sparse convolution network for further feature extraction. The original 3D sparse convolution network used a stride of 2 for three down-sampling operations, directly sending the results to the subsequent network without considering the interaction between multi-scale features. This down-sampling leads to a reduction in feature map resolution, resulting in the loss of some feature information, which is particularly detrimental for detecting small objects.

To enhance the effectiveness of pillarized feature extraction and mitigate information loss, we propose a fine-grained pillar feature extraction module. Specifically, we denote the feature results extracted from each convolution layer as F_{convi}, corresponding to the results of sparse convolution feature extraction from each layer. Thus, the features extracted from the first and second layers are denoted as F_{conv1} and F_{conv2}. Before implementing feature interaction, we first align the features across layers and then use feature concatenation for preliminary feature fusion, enabling feature reuse. Finally, we incorporate sparse channel and spatial attention mechanisms to enhance the contribution of task-relevant channels and strengthen the feature response in the target areas while weakening background interference, achieving further feature fusion.The fused results are then directly fed into the subsequent convolution network. Our execution strategy for the second and third layers, F_{conv2} and F_{conv3} is similar, continuing until the final feature extraction is complete. The pillar feature extraction module not only

effectively enhances the network's ability to express fine-grained local and global features but also compensates for the information loss during the down-sampling process, thereby improving the accuracy of 3D object detection. It also provides an efficient and detailed solution for feature extraction from point cloud data in complex environments.

3.3 Transformer-based Point-Pillar Feature Fusion Module

Existing methods indicate that incorporating transformers into 3D object detection can effectively improve detection accuracy by establishing long-range relationships between voxels through efficient attention operations. Similarly, to leverage various types of information, we propose a transformer-based Point-Pillar Attention Fusion Module, which combines three types of 3D features with attention operations to model the dependencies between different source features.

Specifically, we denote the key point features, pillar features, and BEV (Bird's Eye View)features obtained from the feature extraction phase as F_{point},F_{pillar} and F_{bev} respectively. Since different types of features may have dimension mismatches before fusion, we first perform feature alignment, followed by concatenation to combine F_{point} F_{pillar} and F_{bev}.

$$F_{\text{concat}} = \left[\tilde{F}_{\text{point}} \oplus \tilde{F}_{\text{pillar}} \oplus F_{\text{bev}}\right] \tag{1}$$

Next, we apply a multi-head attention mechanism, projecting the feature vectors into Q/K/V spaces through linear projections.

$$(Q, K, V) = F_{\text{concat}}(W_q, W_k, W_v) \tag{2}$$

Where $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$d×d represents the learnable weights.

The projected feature vectors are split into multiple orthogonal subspaces based on the number of attention heads, allowing each attention head to independently compute attention weights using scaled dot-product attention in different semantic subspaces, thus achieving multi-scale feature fusion.

$$F_{\text{attn}} = \text{Softmax}\left(\frac{QK^{\top}}{\sqrt{d}}\right)V \tag{3}$$

where d is the feature dimension,$\sqrt{d}$ serves as the scaling factor, $QK^{\top}$ computes the pairwise similarity between features, Softmax normalizes the similarity scores to obtain attention weights.

Finally, we know that the attention mechanism excels at capturing relationships between features but lacks the ability to perform complex nonlinear transformations on individual features. To address this, we use a Feedforward Neural Network (FFN) to independently apply nonlinear enhancements to each feature vector generated by the attention mechanism. This part consists of two fully connected layers and an activation function, allowing for more refined processing of the attention output.

$$F_{\text{out}} = \text{LayerNorm}\big(F_{\text{attn}} + \text{FFN}(F_{\text{attn}})\big) \tag{4}$$

As a result, we achieve an overall fusion of point features with pillar and BEV features, effectively enriching the contextual information and providing a more informative feature representation for optimizing target areas in the second stage.

3.4 Proposal Generation and Refinement Network

Since we are using a two-stage network, in the first stage, we have generated a BEV map through the pillarized 3D backbone. The Region Proposal Network based on the BEV can generate 3D proposals with higher recall rates, resulting in the first stage object detection outcomes. The final goal is to optimize the first-stage predicted proposals using the fused features from both point and pillar representations.

For each 3D proposal generated in the first stage, we uniformly sample 216 grid points in a 6×6×6 configuration within its bounding box. For each grid point g_i, we search for nearby key points within a spherical neighborhood of radius $r^{(g)}$ and collect their features.These key point features are concatenated and then aggregated through a shared MLP and max pooling to obtain the feature for the grid point F_{gi}. Finally,the 216 RoI-grid feature vectors are flattened and passed through two layers of MLP (256 dimensions) to generate the global feature representation for the 3D proposal.

Subsequently, using the global features extracted from RoI-grid pooling for each 3D proposal, we construct a confidence prediction branch and a bounding box optimization branch to predict the IoU confidence score of the proposals, as well as to predict the center position, size, and orientation through bounding box regression.

4 Experiments

4.1 Implementation Details

We evaluate our method on the KITTI [19] object detection benchmark.The KITTI benchmark has been widely used for 3-D object detection evaluation. It consists of a training set with 7,481 LiDAR samples and a test set with 7,518 LiDAR samples. The training set is commonly separated into train split (3712 samples) and val split (3769 samples) [20].The dataset primarily contains three categories of objects ,Car, Pedestrian, and Cyclist. Object instances across different classes are further classified into easy, moderate and hard splits.To ensure the fairness of comparative experiments, we strictly adhere to the public data partitioning protocol: the model is trained on the training set and evaluated against other state-of-the-art methods on both the validation set and the online test set. For evaluation metrics, we employ Average Precision (AP), where the validation set performance is calculated using the 40 recall positions to maintain consistent evaluation standards with existing best-performing methods.

For the KITTI dataset, we set the detection range of the point cloud to [0, 70.4] along the X-axis, [-40, 40] along the Y-axis, and [-3, 1] along the Z-axis. The original point cloud input contains approximately 20,000 points per frame, and after sampling, 2048 points are used for input. Data augmentation is performed using random flipping, rotation, and scaling. A voxel size of (0.05m, 0.05m, 0.1m) is used as the pillar input to voxelize each scene.

Our model is trained on a GeForce RTX 4090D GPU. The training process employs the Adam optimizer with a one-cycle learning rate schedule, using 0.01 as the initial learning rate.A total of 80 epochs were trained with a batch size of 2. we set the IOU (Intersection over Union) threshold value of Car to 0.7, and that of Pedestrians and Cyclists to 0.5.

4.2 Comparative Experiment on the KITTI Dataset

The results on the KITTI test set are summarized in Table 1. The table is vertically divided into two sections, corresponding to one-stage methods and two-stage methods. Our model is compared with the current best models in three categories: car, pedestrian, and cyclist. The results demonstrate that our model has competitive accuracy. Specifically, in the Car category, our model outperforms baseline PV-RCNN by an average of 2.89% across three difficulty levels. In the Pedestrian category, our model outperforms PV-RCNN by an average of 31.87% across all three difficulty levels. In the Cyclist category, our model outperforms PV-RCNN by an average of 14.36% across three difficulty levels. Notably, the improvement in Pedestrian detection is relatively large, indicating that our method significantly enhances small object detection performance.

Table. 2. Comparative Evaluation with State-of-the-Art Methods on KITTI test Split.

Method	Types	Car 3D AP			Pedestrian 3D AP			Cyclist 3D AP		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
PointPillars	One	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92
SECOND	One	83.34	72.55	65.82	43.03	35.92	33.56	71.33	52.08	45.83
IA-SSD	One	88.87	80.32	75.10	47.90	41.03	37.98	82.36	66.25	59.70
SVGA-Net	One	87.33	80.47	75.91	48.48	40.39	37.92	78.58	62.28	54.88
PV-RCNN	Two	90.25	81.43	76.82	52.17	43.29	40.29	78.60	63.71	57.65
Part A^2	Two	87.81	78.49	73.51	53.10	43.35	40.06	79.17	63.52	56.93
P2V-RCNN	Two	86.96	81.45	77.20	50.91	43.19	40.81	78.62	63.13	56.81
STD	Two	87.95	79.71	75.09	53.29	42.27	38.35	78.69	61.59	61.42
HPV-RCNN	Two	89.33	80.61	75.53	52.54	43.86	41.56	84.24	69.56	61.42
HybridPillars	Two	90.06	81.48	76.94	-	-	-	81.42	66.05	59.59
Ours	Two	**90.93**	**83.16**	**81.28**	**65.25**	**59.06**	**54.03**	**89.66**	**71.41**	**67.41**

Table. 2. Comparative Evaluation with State-of-the-Art Methods on KITTI val Split.

Method	Types	Car 3D AP			Pedestrian 3D AP			Cyclist 3D AP		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
PointPillars	One	84.01	76.11	72.19	59.45	50.81	44.98	85.10	65.65	60.32
SECOND	One	87.12	79.30	75.91	50.66	47.82	40.54	80.31	64.98	61.01
PVB-SSD	One	90.98	82.06	79.34	61.76	56.55	51.59	82.46	68.67	63.55
PV-SSD	One	88.53	77.80	75.82	-	-	-	84.73	70.17	66.33
IA-SSD	One	90.47	81.72	78.20	55.90	50.53	44.00	90.23	73.25	68.41
SVGA-Net	One	88.93	81.87	79.13	56.06	50.44	43.93	86.16	69.08	62.96
PV-RCNN	Two	91.86	82.85	80.31	59.97	52.37	46.59	86.89	70.64	66.36
Part A^2	Two	90.23	80.45	77.65	58.77	51.89	46.25	85.40	68.82	64.55
M3DETR	Two	90.28	81.73	76.96	45.70	39.94	37.66	83.83	66.74	59.03
PG-RCNN	Two	89.38	82.13	77.33	47.99	41.04	38.71	82.77	67.82	61.25
SCNet3D	Two	89.16	82.35	77.72	51.69	44.64	41.44	82.11	67.55	62.12
Ours	Two	**91.02**	**82.98**	**81.20**	**64.77**	**57.97**	**53.06**	**87.70**	**70.06**	**65.61**

As shown in Tab. 2, we further compare our model with the latest 3D object detection methods on the KITTI validation set across all categories. Compared to baseline PV-RCNN, our method performs better in the Car category at the medium and hard difficulty levels, with a slight decrease in the Easy difficulty level. For the Pedestrian category, our model shows a large improvement across all three difficulty levels. For Cyclist detection, the detection accuracy decreases slightly by 0.35%. This indicates that our method is effective in capturing fine-grained features, which is consistent with the results shown on the test set. Finally, Figure 2 displays the qualitative results of our model HPP-TNet on the KITTI test set.

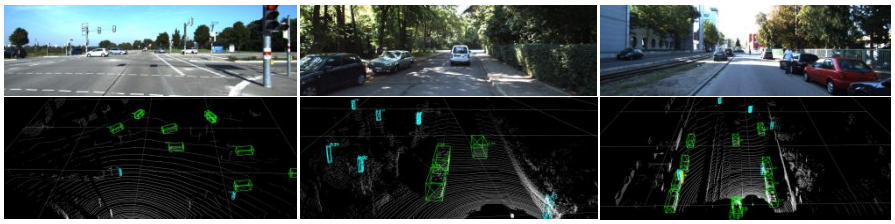

Fig. 2. Qualitative results achieved on the KITTI test set. The first row shows real driving scenes from the KITTI dataset, while the second row exhibits the 3D detection results of HPP-TNet.

4.3 Ablation Experiment on KITTI Dataset

To validate the effectiveness of our proposed CFPEM and TMFFM, we conducted several experiments on the KITTI dataset to evaluate the roles of the Compact Fine-grained Pillar Feature Extraction Module (CFPEM) and the Transformer-based Multi-scale Feature Fusion Module (TMFFM). For a fair and comprehensive evaluation, we calculated the average precision at 40 recall positions and verified the performance improvements of each module under different detection difficulties through controlled variable experiments.

As shown in Table 3, combining the CFPEM and TMFFM modules achieves an effective improvement of 0.5% to 2.7% across difficulty levels. The experiments demonstrate that an optimized feature extraction network, along with enhanced feature fusion, is beneficial for improving the overall detection accuracy in 3D object detection tasks.

Table. 3. Ablation Study of Different Components

CFPEM	TMFFM	3D mAP (%)		
		Easy	Moderate	Hard
√		79.18	70.08	66.77
	√	80.80	70.99	66.89
√	√	81.30	71.47	67.08

5 Conclusion

In this paper, we propose a two-stage 3D object detection framework HPP-TNet that integrates point and pillar features. This method extracts point, multi-scale pillar, and

BEV features within a unified backbone for feature fusion, thereby enhancing the detection accuracy of small targets. We designed the Compact Fine-grained Pillar Feature Extraction Module (CFPEM), which effectively retains fine-grained local features and global context information from the voxel backbone network through a feature reuse strategy, significantly alleviating the feature loss problem during the traditional pillar downsampling process. Furthermore, to fully associate different multi-source features of a point, we applied the multi-head attention mechanism from transformers to the feature fusion module, replacing traditional feature concatenation methods. The fused features are then sent to the candidate generation layer for more accurate 3D result prediction. Experiments on the KITTI dataset show that our proposed algorithm achieves state-of-the-art performance compared to several existing excellent algorithms, with particularly significant improvements in the detection of small targets, such as pedestrians.

However, as a two-stage detection framework, the current model incurs relatively high computational overhead. Future research will focus on developing lightweight feature fusion strategies to reduce computational complexity while maintaining accuracy.

References

1. Qi, C.R., Su,H., Mo, K., et al.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), LNCS, vol. 10112, pp. 652–660. Springer, Heidelberg (2017).
2. Qiao, R., Ji, H., Zhu, Z., et al.: Local-to-global Semantic Learning for Multi-view 3D Object Detection from Point Cloud. IEEE Transactions on Circuits and Systems for Video Technology (2024).
3. Lang, A.H., Vora, S., Caesar, H., et al.: PointPillars: Fast Encoders for Object Detection from Point Clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), LNCS, vol. 11562, pp. 12697–12705(2019).
4. Sun L, Li Y, Qin W. PEPillar: a point-enhanced pillar network for efficient 3Dobject detection in autonomous driving[J]. The Visual Computer, 41(3): 1777-1788(2025)
5. Shi, S., Guo, C., Jiang, L., et al.: PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), LNCS, vol. 12346, pp. 10529–10538(2020).
6. Zhang H, Luo G, Wang X, et al. SASAN: shape-adaptive set abstraction network for point-voxel 3D object detection[J]. IEEE Transactions on Neural Networks and Learning Systems, vol. 36, no. 2, pp. 2465-2479(2023).
7. Qi, C.R., Yi, L., Su, H., et al.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: Advances in Neural Information Processing Systems (NeurIPS), LNCS, vol. 10234, pp. 5099–5108(2017).
8. Shi, S., Wang, X., Li, H.: PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), LNCS, vol. 11234, pp. 770–779(2019).
9. Zhou, Y., Tuzel, O.: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), LNCS, vol.11211, pp. 4490–4499(2018).

10. Yan, Y., Mao, Y., Li, B.: SECOND: Sparsely Embedded Convolutional Detection. In: IEEE International Conference on Intelligent Robots and Systems (IROS), LNCS, vol. 11003, pp. 1–8(2018).
11. Deng, J., Shi, S., Li, P., et al.: Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In: AAAI Conference on Artificial Intelligence, LNCS, vol. 12533, pp. 1201–1209(2021).
12. Noh, J., Lee, S., Ham, B.: HVPR: Hybrid Voxel-Point Representation for Single- Stage 3D Object Detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), LNCS, vol. 12922, pp. 14605–14614. Springer, Cham (2021).
13. Feng, C., Xiang, C., Xie, X., et al.: HPV-RCNN: Hybrid Point-Voxel Two-Stage Network for LiDAR-Based 3D Object Detection. IEEE Transactions on Computational Social Systems 10(6), 3066–3076 (2023).
14. Cao, J.C., Tao, C., Zhang, Z., et al.: Accelerating Point-Voxel Representation of 3D Object Detection for Automatic Driving. IEEE Transactions on Artificial Intelligence 5(1), 254–266 (2023).
15. Li,B., Chen,J., Li,X., et al.: VFL3D: A Single-Stage Fine-Grained Lightweight Point Cloud 3D Object Detection Algorithm Based on Voxels. IEEE Transactions on Intelligent Transportation Systems 25(2), 1123–1135 (2024).
16. Chen, Y., Yang, Z., Zheng, X., et al.: PointFormer: A Dual Perception Attention-Based Net- work for Point Cloud Classification. In: Asian Conference on Computer Vision (ACCV), LNCS, vol. 13842, pp. 3291–3307(2022).
17. Mao, J., Xue, Y., Niu, M., et al.: Voxel Transformer for 3D Object Detection. In: IEEE/CVF International Conference on Computer Vision (ICCV), LNCS, vol. 13682, pp. 3164–3173(2021).
18. Leng, Z., Sun, P., He, T., et al.: PVTransformer: Point-to-Voxel Transformer for Scalable 3D Object Detection. In: IEEE International Conference on Robotics and Automation (ICRA), LNCS, vol. 14218, pp. 4238–4244(2024).
19. Geiger, A., Lenz, P., Stiller, C., et al.: Vision Meets Robotics: The KITTI Dataset. International Journal of Robotics Research 32(11), 1231–1237 (2013).
20. Li, Yidi, et al.: PVAFN: Point-Voxel Attention Fusion Network with Multi- Pooling Enhancing for 3D Object Detection. Expert Systems with Applications (2024).