Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on deep neural network-based multimodal target detection algorithm and its application to point cloud data

Niya Dong[1], Yi Lin[1,✉]

[1] *College of Communication and Information Engineering, Chongqing College of Mobile Communication, Chongqing, 401520, China*

ABSTRACT

Aiming at the problems of poor point cloud data fusion in traditional MLP models, this paper proposes a multimodal 3D target detection network based on KANs. A KANDyVFE encoder incorporating a fusion layer is designed with KANs as the backbone, and a self-attention mechanism is used to dynamically fuse point cloud features. Two datasets, KITTI and WaymoOpen, are selected as 3D target detection datasets to explore the performance level of the algorithm through controlled experiments. Based on ablation experiments, the effectiveness of the KANDyVFE encoder and the self-attention fusion module is verified. The proposed algorithm achieves 80.72% and 80.23% 3DmAP and 3DmAPH on the WaymoOpen dataset for LEVEL_1, which is 2.14% and 2.17% better than the closest BtcDet method, and achieves the same advanced performance on LEVEL_2. When the KANDyVFE encoder module is not used, the 3DmAP and 3DmAPH are only 72.36% and 74.35%, respectively, and the addition of the KANDyVFE encoder and the self-attention fusion module achieves 91.33% and 92.09% for 3DmAP and 3DmAPH, respectively. The experimental results validate the effectiveness of KANs in point cloud applications, and the ablation experiments further demonstrate the performance improvement brought by the designed modules.

Keywords: multimodal target detection; point cloud features; KANs; KANDyVFE; self-attention mechanism

1. Introduction

Three-dimensional target detection is an extremely important part of the perception system. The industry's mainstream three-dimensional target detection using sensors such as LiDAR, optical cameras and other devices, can obtain information about the environment around the vehicle, to identify the road, obstacles, vehicles and pedestrians, etc. [1-2]. Vision sensors using optical cameras can capture images of the scene, capture optical information about the surrounding environment

✉ Corresponding author.
E-mail address: cqcyit@163.com (Y. Lin).
Received 15 January 2024; Revised 25 March 2024; Accepted 30 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-368

through optical lenses, convert it into digital images, and use computer vision technology to achieve detection and localization of targets [3-5]. Vision sensors can provide high-resolution images as well as rich color information, which helps lane line detection, traffic sign recognition, etc [6-7]. However, optical sensors are sensitive to lighting conditions and therefore suffer from strong light, low light, night or shadows, and a single optical camera provides only two-dimensional image information, which does not allow for more three-dimensional information [8-10]. LIDAR utilizes the propagation of a laser beam through the air by measuring the short pulses of the laser beam from emission to reflection to reception time and then calculating the distance between the target and the sensor based on the speed of light [11-12]. The all-round information of the scene is obtained by changing the direction of the laser beam, and a high-precision 3D point cloud map is generated, which contains high-precision distance and position information and is suitable for accurate obstacle detection and localization [13-15]. And because LiDAR is relatively insensitive to light conditions, it can provide reliable data in low light or night scenes [16]. However, LiDAR is costly, bad weather will have an impact on the accuracy, and the large amount of 3D information data puts higher requirements on the computational performance [17-19]. Based on this, exploring more advanced target detection methods in point cloud scenarios is not only of significant value at the theoretical level, but also at the application level, which can improve road safety, promote the development of intelligent transportation, and promote the advancement of autonomous driving technology [20-22].

Recently, the application of artificial intelligence has profoundly affected various fields, and deep learning and multimodal data fusion techniques have significantly promoted the development of the field of automated driving, and a widely used method is to fuse point clouds and visual images for 3D object detection [23]. Visual images provide rich color and texture information, while point clouds accurately reflect the spatial location, geometry, and surface structure information of objects [24-25]. Deep fusion of data from these two modalities improves the accuracy and robustness of 3D target detection, which is of great theoretical significance and practical value for application scenarios such as autonomous driving and robot navigation that require precise perception of the surrounding environment [26-29].

In this paper, firstly, the knowledge related to point cloud data acquired through radar is introduced, and different types of point cloud feature extraction methods are systematically elaborated. A multimodal 3D target detection network based on KANs is proposed to encode the point cloud features and combine the fusion layer based on KANDyVFE. A dynamic weight prediction network based on the attention mechanism is constructed to realize the adaptive fusion of multiple features. The performance of the proposed algorithm is compared with mainstream algorithms in terms of two dimensions: extraction time and extraction accuracy. The contribution of each module to the model performance is verified through ablation experiments to clarify the application effect of the encoding and fusion layers.

2. Design of multimodal target detection algorithm based on KANs

With the rapid development of autonomous driving technology, three-dimensional target detection technology has become a core component of the autonomous driving perception system, and deep learning and multimodal data fusion technology have significantly promoted the development of the autonomous driving field. Although traditional 3D target detection methods have made progress to a certain extent, there are still limitations in feature fusion, multi-scale feature processing, etc., especially the detection performance in complex scenes and different distance scales needs to be improved. To address these challenges, this study proposes a multimodal fusion 3D target detection technique based on Kolmogorov-Arnold Networks (KANs), which realizes efficient processing of point cloud data and accurate target recognition through deep point cloud feature fusion.

2.1. Point cloud data and its characteristics

2.1.1. Point cloud data. Point cloud data is a three-dimensional representation of data that is typically acquired through sensors such as LIDAR. Radar calculates the distance between the target and the radar by transmitting a series of radio waves, waiting for these signals to be reflected by the target and returning, and calculating the distance between the target and the radar based on the time delay of the received reflected signals. The radar also determines the direction of the reflected signals through parameters such as bearing and pitch angle, which allows the radar to determine the position of the target relative to the radar's bearing and altitude. Based on the measured distance and angle information, the radar detections can be converted into point cloud data. The coordinates of each point in the point cloud are determined by its distance, bearing, and pitch angle, so that a three-dimensional shape of the target can be constructed from the point cloud data. However, because of the radar's emission frequency and the influence of parameters such as angle, the point cloud data obtained is often sparse. Taking the KITTI dataset, which is commonly used for autonomous driving, as an example, if the point cloud data is projected onto the corresponding two-dimensional image, only about three percent of the pixel points contain information about the point cloud data.

The point cloud data can be expressed in the form of equation (1).

$$P=\{x_i,y_i,z_i,t_i \mid i=1,2,\ldots,N\} \tag{1}$$

The x_i,y_i,z_i, represents the 3D coordinate information of the i th point, and t_i represents possible other attributes of the i th point, such as color information, intensity information, etc., which makes the point cloud data have a lot of advantages compared with the 2D image data: Point cloud data not only has 3D coordinate information, but also contains some depth information; for outdoor or indoor complex scenes, point cloud data is more effective in dealing with various challenges, such as light changes, occlusion, reflection, etc., so it has a wide range of applications in the field of automatic driving, robot navigation, etc.

2.1.2. Characterization of Point Cloud Data. 3D point cloud data has disorder, rotational invariance and density inhomogeneity.

Point cloud disorder: The disorder of point cloud data refers to the fact that there is no clear order between the points in the point cloud, unlike image data where pixels are arranged in an orderly manner according to a two-dimensional matrix. The acquisition result of point cloud data may be affected by the data acquisition equipment, scanning method or other factors. As a result, the order of point cloud data is usually random or irregular. Points in a point cloud are discrete, and each point has its own 3D coordinates and incidental other attributes, but the order between these points does not affect the point cloud representation. For a point cloud data containing N points, if it is represented as a matrix, there exist a total of $N!$ unequal matrices of combinatorial order. The visualization of point cloud disorder is shown in Figure 1, for a point cloud data $\{p_1,p_2,p_3,p_4,p_5\}$, which is converted into the form of a matrix, this data is read through two different orders, namely $1\to2\to3\to4\to5$ and $2\to4\to3\to1\to5$, and the result is also two matrices with different orders, which proves that the disordered nature of the point cloud brings a This can prove that the disorder of point cloud brings a great challenge to the subsequent processing of feature extraction from point cloud data.

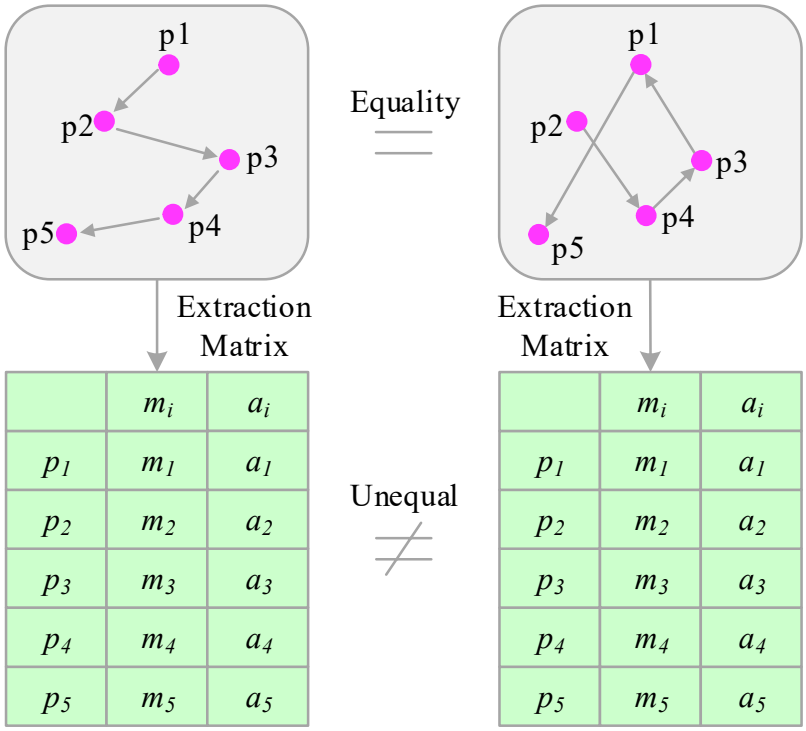


Fig. 1. Disorder of point cloud

Rotational invariance of the point cloud: the rotational operation of the point cloud data in 3D space will not change the geometric structure and features described by the point cloud data. That is to say, if any rotation operation is performed on the point cloud data, its overall shape and structure remain unchanged. For some specific application scenarios, such as point cloud alignment and alignment, it is sometimes necessary to consider the rotation of point cloud data. In these cases, rotational invariance can be used to simplify the processing of the problem and improve the efficiency and robustness of the algorithm.

Density inhomogeneity of point cloud: Unlike image data where pixels are uniformly arranged according to a two-dimensional matrix, the distribution of points in various regions of point cloud data is not uniform, i.e., there is a phenomenon that some regions may be dense while others may be sparse. This non-uniformity can be due to a variety of reasons, including the resolution of the acquisition device, the sampling method, the complexity of the object surface, and occlusion.

2.2. Point cloud feature extraction

The irregularity and disorder characteristics of point clouds bring challenges to point cloud data processing, so these difficulties need to be overcome in point cloud feature extraction. Currently, point cloud feature extraction methods are mainly divided into three categories: voxel-based methods, point-based methods and graph-based methods, and a comparison of different point cloud feature extraction methods is shown in Figure 2.

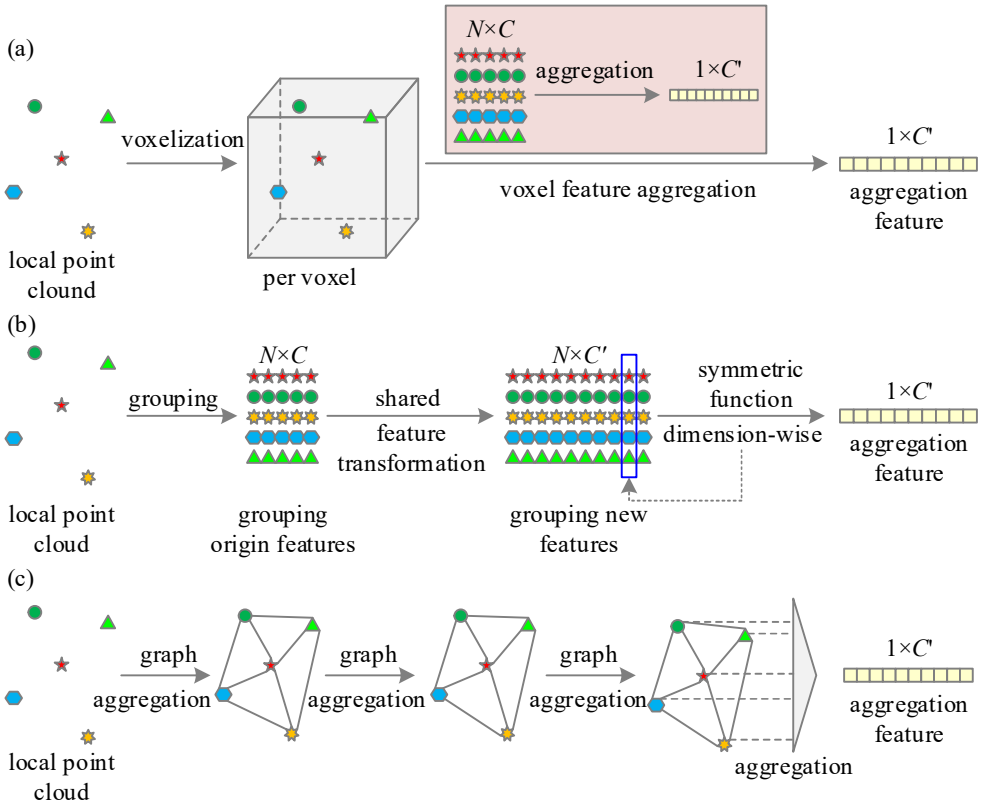


Fig. 2. Comparision of point cloud feature extraction methods

2.2.1. Voxel-based feature extraction. The voxel-based feature extraction method divides the point cloud region into regular 3D voxel rasters and extracts the point cloud features in voxels to generate a regularly structured 4D voxel feature map. For a point cloud P, its range on each axis of the Cartesian coordinate system is $[D,H,W]$, and if the voxel size is set to $[v_D,v_H,v_W]$, the resolution of the raster after voxelization will be $[D',H',W']$, where $D'=D/v_D, H'=H/v_H$, $H'=H/v_H$, and $W'=W/v_W$. As shown in Fig. 2(a), for each voxel, the features $F\in R^{N\times C}$ (R denotes the dimensionality) of the internal point cloud are aggregated using the voxel feature aggregation module, which generates voxel feature vectors $f\in R^{1\times C'}$ with the same sizes, namely

$$f=g_{VFA}(F) \tag{2}$$

Where: g_{VFA} denotes the voxel feature aggregation function, e.g., the average pooling function. The input feature dimension of each voxel depends on the number of local points it contains, but after being processed by the voxel feature aggregation module, each voxel corresponds to a feature vector with the same dimension, which converts the unstructured point cloud P into a regular four-dimensional voxel feature map $V\in R^{D'\times H'\times W\ prime\times C'}$.

2.2.2. Point-based feature extraction. Point-based methods process point clouds directly without additional operations such as voxelization, maximizing the retention of geometric information in the point cloud. For a point cloud P, a series of key points are sampled and used to group the point cloud. As shown in Fig. 2(b), for a group containing N points, the local point cloud is characterized by $F\in R^{N\times C}$, which is first transformed by using the weight-sharing feature transformation function to transform the dimension of the feature into $F'\in R^{N\times C'}$, and then the symmetric function is used to channel-by-channel Aggregate the features of N points to generate the grouped feature vector $f\in R^{1\times C'}$, thus realizing the feature extraction of the point cloud, i.e.

$$f = S\left(H(F)\right) \tag{3}$$

Where: H is the feature transformation function, which can be implemented by multilayer perceptron (MLP) learning; S is the symmetric function, such as the maximum pooling function, average pooling function.

2.2.3. Graph-based feature extraction. There is a certain correlation between point cloud data and graph data in graph theory, and the point cloud can be regarded as nodes in the graph and the graph can be constructed by connecting different nodes so as to utilize the graph-based method for point cloud feature extraction. The graph-based method includes 3 steps of graph construction, graph iteration and graph aggregation, as shown in Fig. 2(c). For a localized point cloud, first a point cloud graph needs to be constructed

$$G = \left(\Phi, E\right) \tag{4}$$

Where: $\mathbf{\Phi}$ is the set of node features, $\varphi_i \in \Phi$ denotes the features of the ith node; E is the set of connectivity features, and $e_{ij} \in E$ denotes the connectivity features of the ith node with the jth node. Then, the features of the graph nodes are gradually updated by iterative optimization, and at the tth graph iteration operation, the node features are updated as

$$\varphi_i^{t+1} = g^t\left(\rho\left(\left\{e_{ij}^t \mid \left(e_{ij}^t \in E\right)\right\}\right), \varphi_i^t\right) \tag{5}$$

Where: ρ is the connection feature aggregation function; g^t indicates that the feature update is realized by using the aggregated connection features and node features. As the number of iterations increases, the sensory field of the node expands, and after T iterations, each node can learn rich features. Finally, the graph aggregation module is utilized to complete the aggregation of node features to obtain the feature representation of the point cloud.

2.3. Point cloud feature fusion based on KANs

In this paper, a multimodal 3D target detection network based on KANs is proposed. The network adopts KANs as the backbone and combines with a fusion layer to construct the voxel encoder KANDyVFE. KANDyVFE realizes a finer cross-modal fusion by encoding the point cloud and dynamically adjusting the weights of the modal features using the fusion layer, which ultimately improves the accuracy of 3D target detection effectively.

2.3.1. KANDyVFE-based point cloud feature encoding. VoxelNet voxelizes the point cloud and proposes a VFE module that converts the points within each voxel into a uniform feature representation. The advantage of this method is the ability to apply 3D convolution directly to the voxelized features. Inspired by this work, we propose KANDyVFE to encode point cloud features. We dynamically voxelize all point clouds to obtain the 3D voxel coordinates of each point. The 7-dimensional features of the point cloud with 3D voxel coordinates and image features are fed into the KANDyVFE layer as input.

In addition, the deviation of each point from the center of the voxel and the deviation of each point from the center of the voxel clustering were calculated. The deviation of each point from the voxel center is defined as:

$$\Delta_{center} = p_i - c_{center} = \left(x_i - x_j, y_i - y_j, z_i - z_j\right) \tag{6}$$

where p_i denotes the 3D coordinates of the i th point cloud and c_{center} denotes the center coordinates of the voxel to which the point cloud belongs. The deviation of each point from the voxel clustering center is defined as:

$$\Delta_{cluster} = p_i - c_{cluster} = \left(x_i - x_j, y_i - y_j, z_i - z_j\right) \tag{7}$$

$$c_{cluster} = \frac{1}{N}\sum_{i=1}^{N} p_i = \left(\frac{1}{N}\sum_{i=1}^{N} x_i, \frac{1}{N}\sum_{i=1}^{N} y_i, \frac{1}{N}\sum_{i=1}^{N} z_i\right) \tag{8}$$

where $c_{cluster}$ denotes the cluster center coordinates of the voxel. Ultimately, the point cloud is characterized as: $\left(x, y, z, \text{int}\, ensity, r, g, b, \Delta_{center}, \Delta_{duster}\right)$.

Past methods have used MLPs to process point clouds, which use fixed activation functions at the nodes.The main difference between KANs and MLPs is the use of learnable activation functions instead of fixed linear weights, which allows KANs to learn patterns dynamically and improve performance with fewer parameters.

$$MLP(x) = \left(W_{N-1} \circ \delta \circ W_{N-2} \circ \cdots \circ W_1 \circ \delta \circ W_0\right) x \tag{9}$$

where W is a linear combination of weight matrices, x represents the inputs, and δ represents the nonlinear function ReLU, i.e., $f(x) = \max(0, x)$.

$$KAN(x) = \left(\Phi_{N-1} \circ \delta \circ \Phi_{N-2} \circ \cdots \circ \Phi_1 \circ \delta \circ \Phi_0\right) x \tag{10}$$

where Φ_i denotes the ith KAN layer. The core difference between the two is the processing mechanism of the network nodes. In MLP, each node first performs a linear transformation and subsequently applies a nonlinear activation function, whereas in KAN, the nodes do not introduce additional nonlinear transformations when aggregating the function outputs. The innovation of KAN is to replace the weight parameters in traditional neural networks with univariate functions (usually in the form of spline functions). This design allows the KAN layer to be more expressive and to model complex functional relationships efficiently with fewer parameters.

The number of parameters in a layer of MLP is:

$$Parameters = d_{in} \times d_{out} + d_{out} \tag{11}$$

d_{in} and d_{out} are the input and output dimensions, respectively.

And the number of parameters of a layer of KAN is:

$$Parameters = d_{in} \times d_{out} \times \left(G + K + 3\right) + d_{out} \tag{12}$$

K is the order of the spline function and G is the number of control points. It can be seen that because of the computation of the design spline function, the KAN has more parameters than the MLP.

The KANDyVFE layer structure is shown in Fig. 3, where the point cloud features are processed through the KANs and converted into higher dimensional representations. Each KAN layer is followed by a BN layer and a ReLU layer.The high-dimensional point features output from the KANs are subsequently fed into the fusion layer along with the image features to obtain fused features. The fusion layer is described in detail in the following subsections. Next, the fused point features are clustered into voxel features by dynamic scattering. In this paper, the maximum value is used for dimensionality reduction:

$$f_k = \max\left(\left\{f_i \mid p_i \in v_k\right\}\right) \tag{13}$$

where v_k is the kth voxel and f_k is the feature of the voxel. p_i is the ith point and f_i is the feature of the corresponding point.

Finally, the fused point features and voxel features are stitched together. The input point cloud feature data is converted into high-dimensional features by stacking the KANDyVFE layers in the proposed voxel encoder.

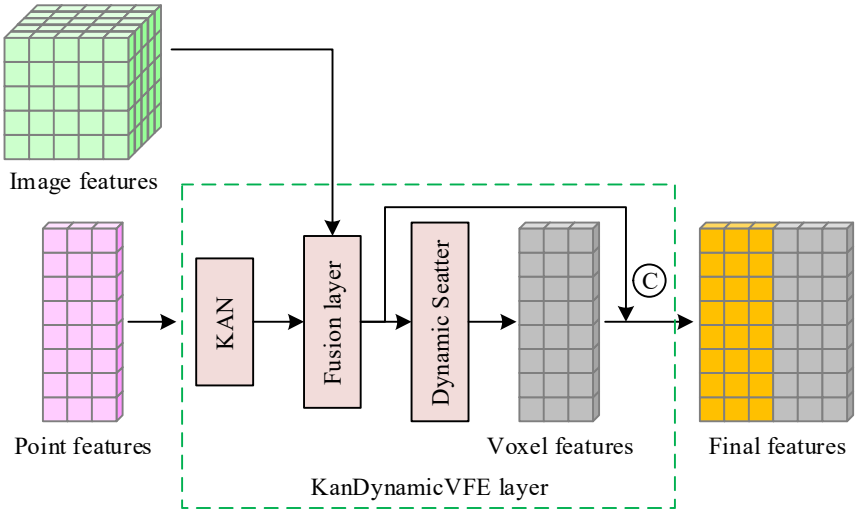


Fig. 3. KANDyVFE layer

2.3.2. Point cloud feature fusion based on attention images. The structure of the fusion layer is shown in Fig. 4, with the image and point features as inputs. First, pixel features corresponding to the pixel coordinates of each point mapped to the image are extracted to ensure that their number matches the point features, while unused features are discarded. Next, the mapped image and point features are transformed through the KAN layer to unify their dimensions for subsequent processing. During the experiments, it was found that directly merging image features and point features may not yield optimal results. Therefore, a self-attention mechanism was used to generate weights from image features and reweight image features with these weights. This method dynamically adjusts the contribution of image features. Finally, the weighted image features and point features are stitched together to generate the final fused features.

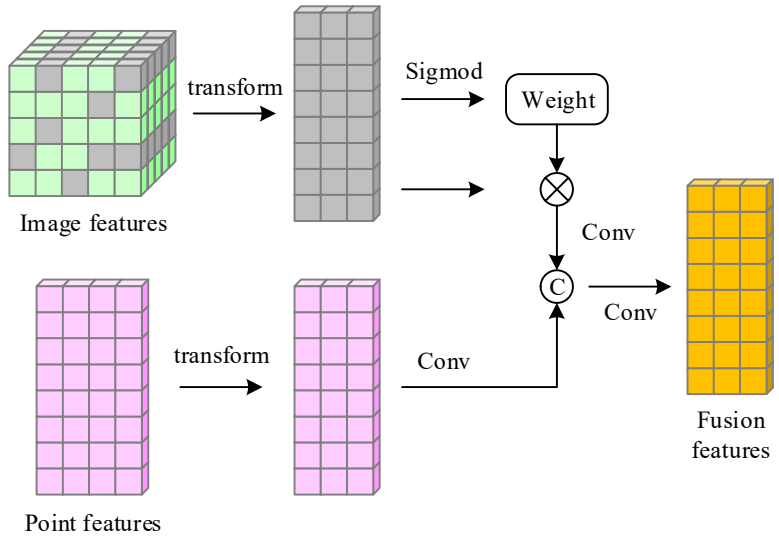


Fig. 4. Attentive image point fusion

3. Evaluation of the effectiveness of multimodal target detection algorithms based on KANs

3.1. Experimental data and evaluation indicators

In order to verify the effectiveness of the proposed algorithm, two widely used datasets, KITTI and WaymoOpen, are selected as the 3D target detection dataset, and the experimental results are compared with other methods. These two datasets, as the mainstream datasets in the field of autonomous driving research, provide rich 3D point cloud data, as well as image, radar, and other sensor data, which provide a multi-dimensional and cross-scenario research basis for researchers. The experimental runtime environment uses MATLAB2021b simulation software as the test platform and Python 3.8 as the main programming language.

In the target detection task, common metrics for evaluating model performance include average precision (AP) and mean average precision (mAP). These metrics provide a comprehensive picture of the model's performance in terms of predictive accuracy and recall.

AP measures the model's performance on a single category, specifically the average of precision rates at different recall levels. Precision is the proportion of positive samples correctly predicted by the model over all samples predicted as positive, and recall is the proportion of positive samples correctly predicted by the model over all actual positive samples. The formula is calculated as follows:

$$\text{Precision rate}\left(Precision\right): P = \frac{TP}{TP + FP} \tag{14}$$

$$\text{Recall rate}\left(\text{Re}\,call\right): R = \frac{TP}{TP + FN} \tag{15}$$

Where TP is the number of true cases, FP is the number of false positive cases, and FN is the number of false negative cases. Under different thresholds, the model will have different precision and recall rates, and the $P-R$ curve drawn can visualize the model performance.

AP is the area under this $P-R$ curve, the value domain is $[0,1]$, the higher value means the better model performance. The formula is expressed as an integral or approximate summation of the area under the $P-R$ curve.

The mAP is the average of the AP values of multiple categories and is used to measure the overall performance of the model on all categories, especially in detection tasks with multiple target categories, and is calculated as follows:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{16}$$

where AP_i is the average accuracy of the ith category and N is the total number of categories.

3.2. Comparative Experimental Analysis

The method of this paper is compared with mainstream 3D target detection algorithms. Among them, PointPillars is a point cloud-based 3D target detection method, which represents the point cloud data by splitting it into voxel-like voxels, and then utilizes convolutional neural networks to extract features and perform target detection. Voxel R-CNN is a voxel-based 3D target detection method, which first converts the point cloud data into voxel representations, and then utilizes 3D convolutional neural networks to perform feature extraction and target detection, which can achieve better performance when dealing with sparse point cloud data. SECOND is another voxel-based 3D target detection method, which uses the network structure of VoxelNet, and can efficiently deal with the point cloud data when performing feature extraction and target detection. PV-RCNN is a 3D target detection method based on the projected view, which projects the point cloud to the different views and combines 2D and 3D information for target detection, which improves the accuracy and robustness of

detection. VoTr is a Transformer-based 3D target detection method, which converts the point cloud data into serialized representations and utilizes the Transformer model for feature extraction and target detection with better detection performance. BtcDet is a deep learning based 3D target detection method that uses the Bottom-up Top-down Cluster (Btx) strategy to detect targets by first clustering from the bottom and then clustering upwards, which is able to effectively deal with irregularly shaped targets.

3.2.1. Extraction time. Ten test samples are selected to explore the variation of point cloud data extraction time of 3D virtual reality object surface image of each algorithm. The comparison of the extraction time of different algorithms is shown in Table 1, from which it can be seen that the average extraction time of point cloud data of the proposed algorithm is 0.21s, and the average extraction time of other algorithms is more than 0.25s, which is higher than that of the proposed algorithm. The comparison fully proves that the proposed method can complete the point cloud data extraction with high efficiency.

Table 1. Comparison of extraction times for different algorithms

Test sample number	PointPillars	Voxel R-CNN	SECOND	PV-RCNN	VoTr	BtcDet	KANs
1	0.34	0.31	0.29	0.27	0.28	0.25	0.22
2	0.31	0.32	0.30	0.28	0.29	0.26	0.19
3	0.28	0.25	0.28	0.27	0.27	0.26	0.18
4	0.32	0.26	0.25	0.27	0.32	0.27	0.23
5	0.29	0.27	0.28	0.29	0.30	0.24	0.24
6	0.27	0.26	0.29	0.30	0.29	0.28	0.21
7	0.27	0.31	0.31	0.28	0.31	0.22	0.18
8	0.28	0.27	0.31	0.27	0.29	0.27	0.19
9	0.29	0.25	0.28	0.26	0.28	0.26	0.22
10	0.31	0.28	0.27	0.27	0.31	0.26	0.24

3.2.2. Extraction accuracy. On the WaymoOpen dataset, the accuracy comparison results of different algorithms are shown in Fig. 5, and the algorithm in this paper achieves the best performance on both 3DmAP and 3DmAPH for LEVEL_1 and LEVEL_2. 3DmAP is a comprehensive evaluation metric to measure the average accuracy of the detection algorithms on different categories, and the final average is taken to obtain 3DmAP. 3DmAPH is based on the calculation of 3DmAP with fuller consideration of the height of the detection frame. The results show that the algorithm in this paper achieves 80.72% and 80.23% on 3DmAP and 3DmAPH at LEVEL_1, which are 2.14% and 2.17% better than the closest BtcDet method, respectively. On 3DmAP and 3DmAPH at LEVEL_2, the algorithm in this paper also achieves advanced performance, which is 5.29% and 5.07% ahead of the closest BtcDet method, respectively.

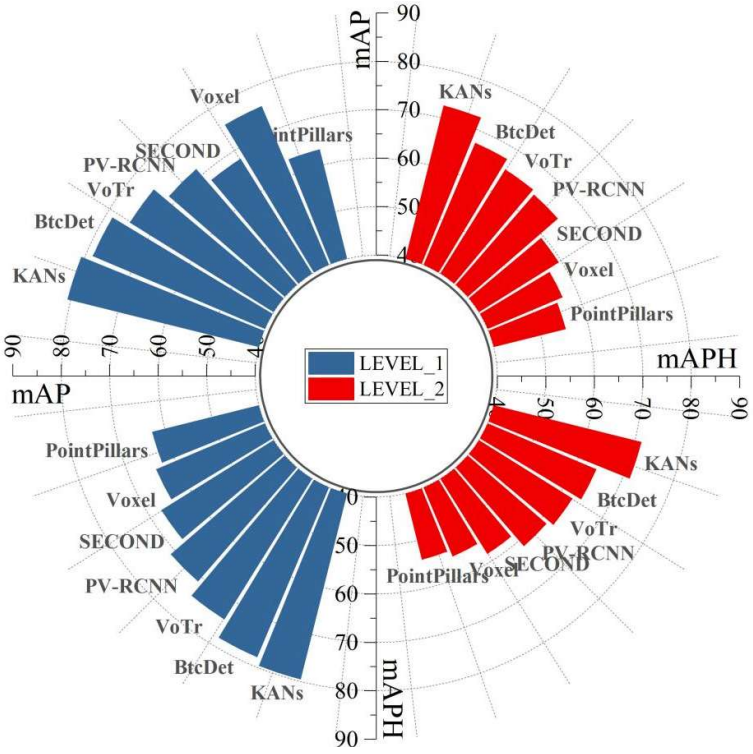


Fig. 5. Comparison results of accuracy of different algorithms

In real-world scenarios, target objects may be in different distance ranges. The evaluation of target detection performance at different distances can help us understand how the target detection algorithms perform at different distances. First, target objects at different distances may have different scales and resolutions, which will affect the representation of the target in the point cloud data and the detection difficulty. Targets at close distances may have higher point cloud densities and richer detail information, while targets at long distances may only be represented with fewer points and have lower resolution. Second, targets at different distances may be subject to different environmental factors and noise interference. Evaluating the target detection performance at different distances can understand the robustness and stability of the algorithm under different environmental conditions. In this paper, the detection performance under different distances is experimented and the results are shown in Fig. 6. This paper's algorithm performs well in both 3DmAP and 3DmAPH for LEVEL_1 in different distance ranges, especially in the distance ranges of 0-30m and 30-50m, this paper's algorithm achieves 97.59% and 78.12% of 3DmAPH, which is much higher than other algorithms. The above results show that this paper's algorithm is able to accurately detect targets at both close and medium distances, with strong adaptability and generalization ability.

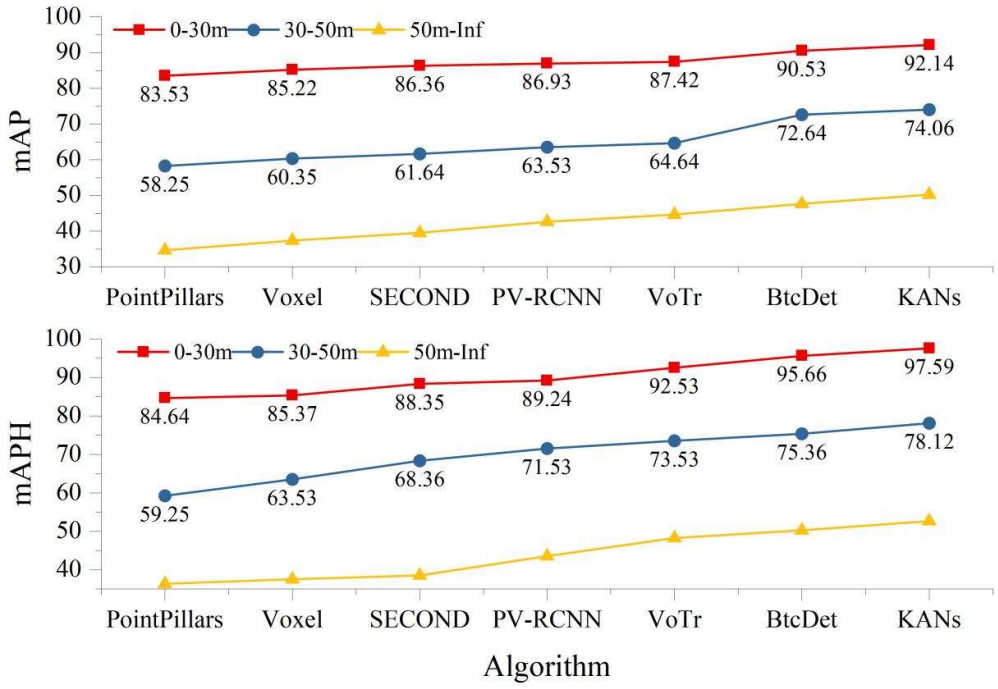


Fig. 6. Precision comparison results of different algorithms at different distances

A comparison of the performance of each algorithm on the KITTI dataset for car, pedestrian and cyclist detection under different difficulty levels is shown in Table 2. The algorithms in this paper achieve excellent performance under all difficulty levels, outperforming other methods, and especially achieve significant improvement in Hard level. First, the results of vehicle detection are analyzed. Under Easy difficulty level, this paper's algorithm achieves 93.08% detection accuracy, and under Mod and Hard difficulty levels, this paper's algorithm achieves 83.97% and 80.93% accuracy respectively, which is much higher than other algorithms. For pedestrian detection, under Easy difficulty level, this paper's algorithm achieves 56.87% accuracy, which is 1.51% higher than the 55.36% of the next best PointPillars. Under Mod and Hard difficulty levels, this paper's algorithm also achieves 45.12% and 43.36% accuracy, respectively, which is significantly higher than other algorithms. For cyclist detection, this paper's algorithm achieves 82.39%, 69.77%, and 59.16% accuracy under Easy, Mod, and Hard difficulty levels, respectively, which are significantly higher than other algorithms.

Table 2. Compares the accuracy results of different algorithms(%)

Algorithm	Cars			Pedestrians			Cyclists		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
PointPillars	82.53	72.64	66.47	55.36	42.75	39.06	73.53	59.25	54.36
Voxel R-CNN	87.03	79.53	75.46	50.35	48.30	40.22	75.04	59.36	55.11
SECOND	88.47	80.55	79.41	53.29	44.24	41.28	78.39	66.37	57.35
PV-RCNN	88.22	81.04	79.33	52.46	40.36	39.27	81.05	62.35	57.44
VoTr	87.48	82.48	78.94	51.08	41.48	42.66	77.48	63.52	55.41
BtcDet	90.42	82.44	76.39	53.09	42.83	41.09	76.22	66.75	56.03
KANs	93.08	83.97	80.93	56.87	45.12	43.36	82.39	69.77	59.16

In summary, compared with the single feature extraction method, multi-feature fusion can capture the shape, structure and context information of the target more comprehensively, which makes the accuracy of detection improved. In addition, the attention mechanism is introduced into the network structure of this paper's algorithm to weight the fusion of features, and the attention mechanism can dynamically adjust the weights of different features according to their importance, so that the network

can pay more attention to the important features and inhibit the unimportant features, which improves the network's characterization and generalization ability, and also improves the accuracy and robustness of detection.

3.3. Analysis of ablation experiments

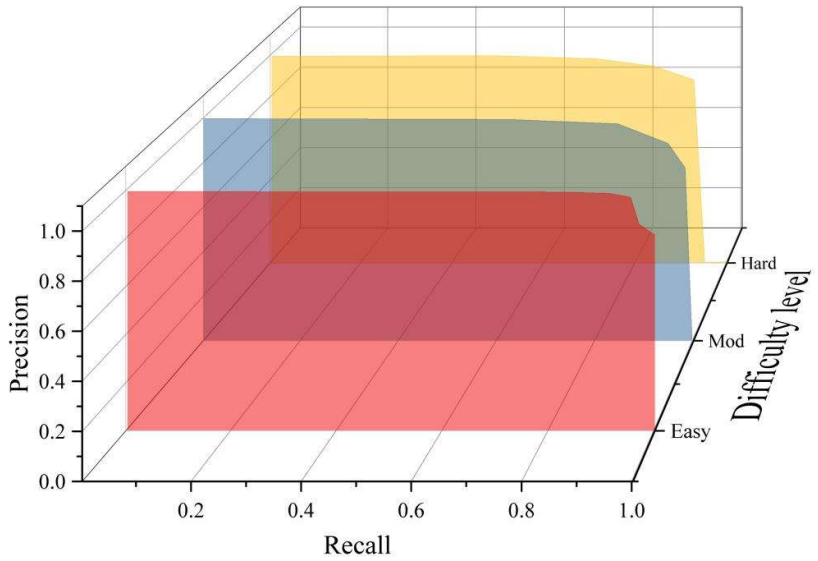
3.3.1. Extraction accuracy. To further validate the effectiveness of different modules, this paper conducts ablation experiments on the WaymoOpen dataset to validate the effectiveness of the KANDyVFE encoder module and the self-attention fusion module, respectively.

First, for the KANDyVFE encoder module, this experiment tests removing the KANDyVFE encoder module and the self-attention fusion module, and the results of the ablation experiment are shown in Table 3. According to Table 3, it can be seen that when the KANDyVFE encoder module is not used, the 3DmAP and 3DmAPH are only 72.36% and 74.35%, respectively, and the accuracy is significantly improved by adding the KANDyVFE encoder module, while the self-attention fusion module further improves the resultant accuracy, and the final 3DmAP and 3DmAPH reach 91.33% and 92.09%, respectively, which fully validates the effectiveness of the KANDyVFE encoder module and the self-attention fusion module.

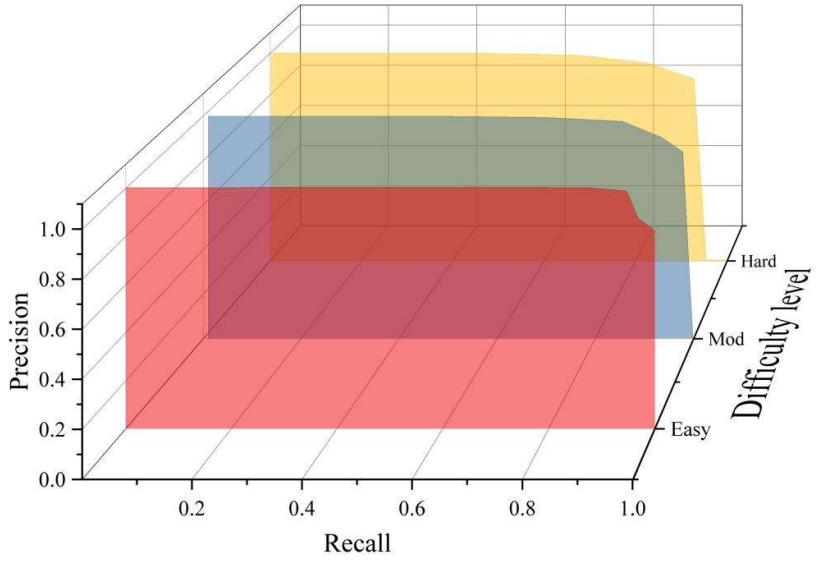
Table 3. Ablation results of KANDyVFE encoder module(%)

Module	Number of point clouds	3DmAP	3DmAPH
None	1024	72.36	74.35
KANDyVFE	1024	90.17	91.28
KANDyVFE+Self-attention fusion	1024	91.33	92.09

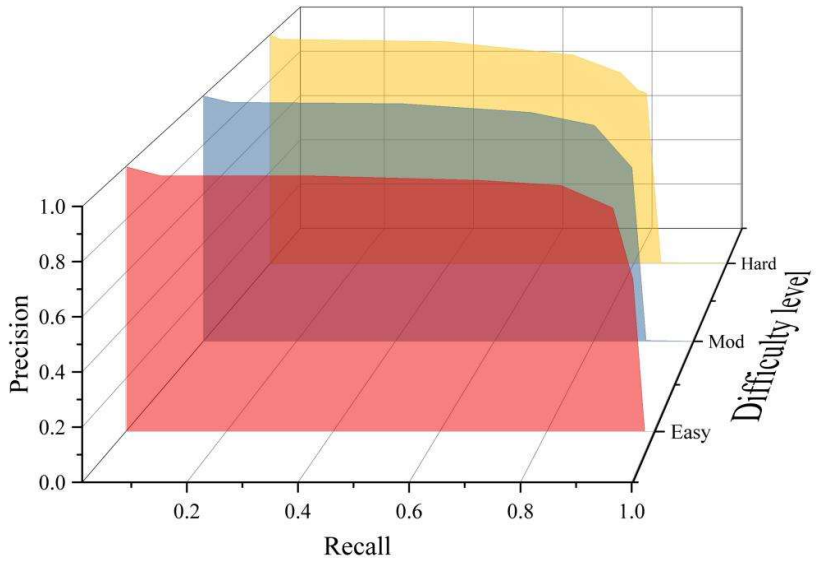
On the KITTI dataset, the algorithm performance is evaluated in four dimensions: accuracy of 3D detection frame, accuracy of detection frame in BEV view, accuracy of 2D detection frame, and accuracy of detecting target rotation angle. To be fair, the experiment adopts the official evaluation criteria, specifically, the IoU threshold is set to 0.7 for the detection of cars and 0.5 for the detection of pedestrians, and the detection accuracy is calculated uniformly with 40 sample point recall. The Recall-Precision curves are plotted based on the detection results as shown in Fig. 7. Fig. 7(a-d) shows the detection results of 2D detection frame, detection target rotation angle, 3D detection frame, and BEV view, respectively, and the accuracy of the four dimensions reaches 91.87%, 91.02%, 76.83%, and 85.01% respectively under the Hard difficulty level.



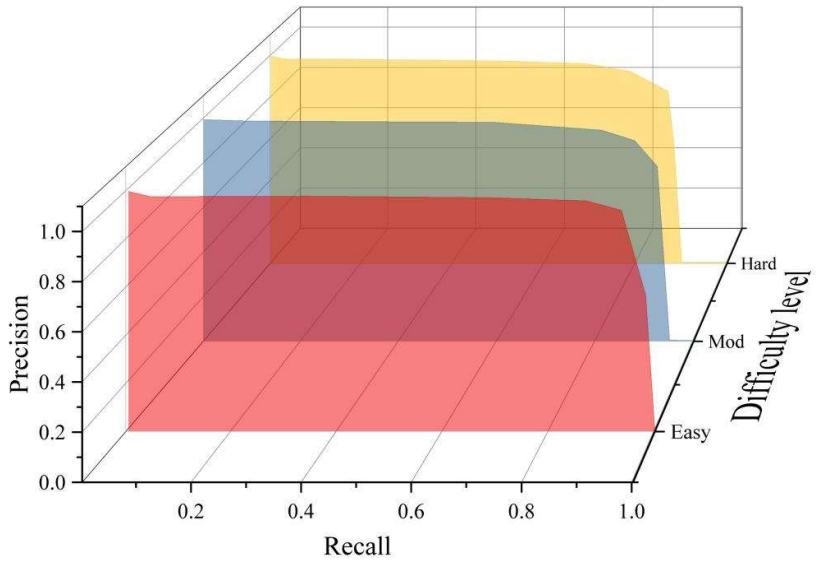
(a)2D Detection



(b)Orientation



(c)3D Detection



(d)BEV

Fig. 7. Recall-Preecision curve

3.3.2. Training efficiency. The mainstream coding modules are compared with KANDyVFE, and the comparison results of inference time and training speed of each module are shown in Table 4, FPS denotes the number of images processed per second, Slowest denotes the training time of the slowest epoch, Fastest denotes the training time of the fastest epoch, Average denotes the average training time of all the epochs, Std denotes the standard deviation of all training times. It can be seen that KANDyVFE has a significant improvement in training time, and compared with the sub-optimal MFF, the average training time is reduced by 0.0501s, and the FPS is improved by 1.8. The standard deviation of KANDyVFE is only 0.0075, which can be seen that the distribution of the training time tends to be more balanced, and the model tends to be stable.

Table 4. Comparison results of reasoning time and training speed

Component	FPS (img/s)	Time (s)			Std
		Slowest	Fastest	Average	
Baseline	5.2	1.2085	1.1944	1.1987	0.0169
GPHE	5.6	0.8868	0.7565	0.8254	0.0553
PGHE	6.3	0.7965	0.7688	0.7756	0.0428
RPGHE	6.6	0.6886	0.6525	0.6675	0.0386
MPF	7.2	0.6843	0.6434	0.6789	0.0243
MFF	7.9	0.6017	0.5867	0.5922	0.0139
KANDyVFE	9.7	0.5501	0.5399	0.5421	0.0075

3.3.3. Integration strategies. The effect of this paper's algorithm combined with different fusion strategies on the detection results on the KITTI dataset is shown in Table 5. It can be seen that the mAP of this paper's fusion strategy is higher than that of other methods by 6.45%, 7.49%, 7.58% and 8.64%, respectively. This indicates that the fusion strategy of this paper has obvious advantages over other methods.

Table 5. Influences of different fusion strategies on detection results(%)

Component	3D			mAP
	Cars	Pedestrians	Cyclists	
Self-attention fusion	76.35	49.04	55.18	60.19
DFF	73.66	41.28	46.27	53.74
Transfusion	71.20	36.22	50.68	52.70
3D-CVF	71.07	38.54	48.23	52.61
CLOCs	70.39	37.67	46.58	51.55

4. Conclusion

In this paper, we design the effect of multimodal target detection algorithm based on KANs and experimentally explore its effectiveness in point cloud data.

The average extraction time of point cloud data for the proposed algorithm is 0.21s, and the 3DmAP and 3DmAPH of LEVEL_1 on WaymoOpen dataset reach 80.72% and 80.23%, which are 2.14% and 2.17% better than the closest BtcDet method, respectively, and the same advanced performance is achieved on LEVEL_2. Both 3DmAP and 3DmAPH of LEVEL_1 perform well in different distance ranges, especially in the distance ranges of 0-30m and 30-50m, it achieves 97.59% and 78.12% 3DmAPH respectively, which is much higher than other algorithms. The above results show that the algorithm in this paper is able to accurately detect targets at both close and medium distances with strong adaptability and generalization ability. On the KITTI dataset, this paper's algorithm achieves excellent performance under all difficulty levels, surpassing other methods, especially achieving significant improvement on the Hard level, with accuracies of 80.93%, 43.36%, and 59.16%, respectively.

Ablation experiments were conducted on the WaymoOpen dataset, when the KANDyVFE encoder module was not used, the 3DmAP and 3DmAPH were only 72.36% and 74.35%, respectively, and the addition of the KANDyVFE encoder and self-attention fusion module reached 91.33% and 92.09% for 3DmAP and 3DmAPH, respectively, which fully verifies that the effectiveness of the KANDyVFE encoder module and self-attention fusion module. Comparing the mainstream encoding module with KANDyVFE, KANDyVFE reduces the average training time by 0.0501s and improves the FPS by 1.8 compared with the sub-optimal MFF.Comparing the mainstream fusion strategy with the fusion

strategy of this paper, the mAP of this paper's fusion strategy is higher than that of the other methods by 6.45%, 7.49%, 7.58% and 8.64% respectively.

References

[1] Ghasemieh, A., & Kashef, R. (2022). 3D object detection for autonomous driving: Methods, models, sensors, data, and challenges. Transportation Engineering, 8, 100115.

[2] Qian, R., Lai, X., & Li, X. (2022). 3D object detection for autonomous driving: A survey. Pattern Recognition, 130, 108796.

[3] Meyer, G. P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., & Wellington, C. K. (2019). Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12677-12686).

[4] Arnold, E., Dianati, M., de Temple, R., & Fallah, S. (2020). Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors. IEEE Transactions on Intelligent Transportation Systems, 23(3), 1852-1864.

[5] Hu, P., Ziglar, J., Held, D., & Ramanan, D. (2020). What you see is what you get: Exploiting visibility for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11001-11009).

[6] Khatab, E., Onsy, A., & Abouelfarag, A. (2022). Evaluation of 3D vulnerable objects' detection using a multi-sensors system for autonomous vehicles. Sensors, 22(4), 1663.

[7] Yang, L., Tang, T., Li, J., Yuan, K., Wu, K., Chen, P., ... & Yu, K. (2025). BEVHeight++: Toward robust visual centric 3D object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[8] Li, J., Liu, Z., Li, L., Lin, J., Yao, J., & Tu, J. (2023). Multi-view convolutional vision transformer for 3D object recognition. Journal of Visual Communication and Image Representation, 95, 103906.

[9] Lentsch, T., Caesar, H., & Gavrila, D. (2024). Union: Unsupervised 3d object detection using object appearance-based pseudo-classes. Advances in Neural Information Processing Systems, 37, 22028-22046.

[10] Usmani, K., O'Connor, T., Wani, P., & Javidi, B. (2022). 3D object detection through fog and occlusion: passive integral imaging vs active (LiDAR) sensing. Optics Express, 31(1), 479-491.

[11] Wu, Y., Wang, Y., Zhang, S., & Ogai, H. (2020). Deep 3D object detection networks using LiDAR data: A review. IEEE Sensors Journal, 21(2), 1152-1171.

[12] Chen, C., Fragonara, L. Z., & Tsourdos, A. (2021). RoIFusion: 3D object detection from LiDAR and vision. IEEE Access, 9, 51710-51721.

[13] Alaba, S. Y., & Ball, J. E. (2022). A survey on deep-learning-based lidar 3d object detection for autonomous driving. Sensors, 22(24), 9577.

[14] Zamanakos, G., Tsochatzidis, L., Amanatiadis, A., & Pratikakis, I. (2021). A comprehensive survey of LIDAR-based 3D object detection methods with deep learning for autonomous driving. Computers & Graphics, 99, 153-181.

[15] Li, X., Du, S., Li, G., & Li, H. (2019). Integrate point-cloud segmentation with 3D LiDAR scan-matching for mobile robot localization and mapping. Sensors, 20(1), 237.

[16] Li, M., Rottensteiner, F., & Heipke, C. (2019). Modelling of buildings from aerial LiDAR point clouds using TINs and label maps. ISPRS Journal of Photogrammetry and Remote Sensing, 154, 127-138.

[17] Cho, S., Kim, C., Park, J., Sunwoo, M., & Jo, K. (2020). Semantic point cloud mapping of LiDAR based on probabilistic uncertainty modeling for autonomous driving. Sensors, 20(20), 5900.

[18] Kim, C., Cho, S., Sunwoo, M., Resende, P., Bradaï, B., & Jo, K. (2021). Updating point cloud layer of high definition (hd) map based on crowd-sourcing of multiple vehicles installed lidar. IEEE Access, 9, 8028-8046.

[19] Sobczak, Ł., Filus, K., Domański, A., & Domańska, J. (2021). Lidar point cloud generation for slam algorithm evaluation. Sensors, 21(10), 3313.

[20] He, C., Zeng, H., Huang, J., Hua, X. S., & Zhang, L. (2020). Structure aware single-stage 3d object detection from point cloud. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11873-11882).

[21] Imad, M., Doukhi, O., & Lee, D. J. (2021). Transfer learning based semantic segmentation for 3D object detection from point cloud. Sensors, 21(12), 3964.

[22] Li, Z., Yao, Y., Quan, Z., Xie, J., & Yang, W. (2022). Spatial information enhancement network for 3D object detection from point cloud. Pattern Recognition, 128, 108684.

[23] Wang, Y., Mao, Q., Zhu, H., Deng, J., Zhang, Y., Ji, J., ... & Zhang, Y. (2023). Multi-modal 3d object detection in autonomous driving: a survey. International Journal of Computer Vision, 131(8), 2122-2152.

[24] Liang, M., Yang, B., Wang, S., & Urtasun, R. (2018). Deep continuous fusion for multi-sensor 3d object detection. In Proceedings of the European conference on computer vision (ECCV) (pp. 641-656).

[25] Arikumar, K. S., Deepak Kumar, A., Gadekallu, T. R., Prathiba, S. B., & Tamilarasi, K. (2022). Real-time 3D object detection and classification in autonomous driving environment using 3D LiDAR and camera sensors. Electronics, 11(24), 4203.

[26] Liu, M., Chen, Y., Xie, J., Zhu, Y., Zhang, Y., Yao, L., ... & Zhou, J. T. (2024). MENet: Multi-modal mapping enhancement network for 3D object detection in autonomous driving. IEEE Transactions on Intelligent Transportation Systems.

[27] Li, Y., Zeng, K., & Shen, T. (2023). CenterTransFuser: radar point cloud and visual information fusion for 3D object detection. EURASIP Journal on Advances in Signal Processing, 2023(1), 7.

[28] Chen, Q., Liu, Z., Zhang, Y., Fu, K., Zhao, Q., & Du, H. (2021, May). RGB-D salient object detection via 3D convolutional neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 2, pp. 1063-1071).

[29] Gao, M., Jiang, J., Zou, G., John, V., & Liu, Z. (2019). RGB-D-based object recognition using multimodal convolutional neural networks: A survey. IEEE access, 7, 43110-43136.