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

Fine-Grained Pillar Feature Encoding Via Spatio-Temporal Virtual Grid for 3D Object Detection

Konyul Park[1*], Yecheol Kim[2*], Junho Koh[2], and Byungwoo Park[1] and Jun Won Choi[3†]

***Abstract*— Developing high-performance, real-time architectures for LiDAR-based 3D object detectors is essential for the successful commercialization of autonomous vehicles. Pillar-based methods stand out as a practical choice for onboard deployment due to their computational efficiency. However, despite their efficiency, these methods can sometimes underperform compared to alternative point encoding techniques such as Voxel-encoding or PointNet++. We argue that current pillar-based methods have not sufficiently captured the fine-grained distributions of LiDAR points within each pillar structure. Consequently, there exists considerable room for improvement in pillar feature encoding. In this paper, we introduce a novel pillar encoding architecture referred to as *Fine-Grained Pillar Feature Encoding* (FG-PFE). FG-PFE utilizes *Spatio-Temporal Virtual* (STV) grids to capture the distribution of point clouds within each pillar across vertical, temporal, and horizontal dimensions. Through STV grids, points within each pillar are individually encoded using Vertical PFE (V-PFE), Temporal PFE (T-PFE), and Horizontal PFE (H-PFE). These encoded features are then aggregated through an *Attentive Pillar Aggregation* method. Our experiments conducted on the nuScenes dataset demonstrate that FG-PFE achieves significant performance improvements over baseline models such as PointPillar, CenterPoint-Pillar, and PillarNet, with only a minor increase in computational overhead.**

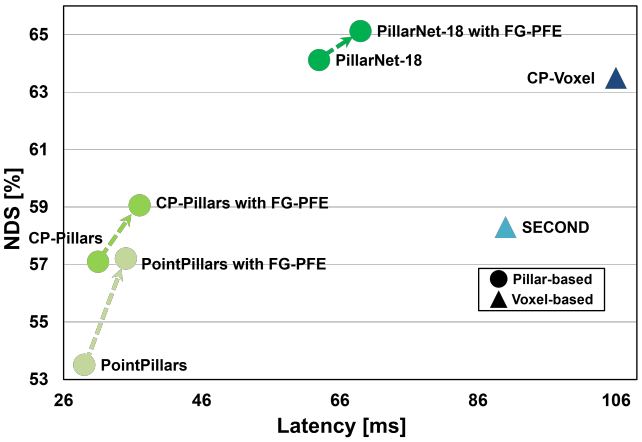

Fig. 1. **Performance versus latency of several 3D object detectors evaluated on nuScenes *val* split**: CP-Pillars denotes CenterPoint [1] with PointPillars backbone. Latency is measured with a single NVIDIA TITAN RTX GPU. When incorporated into various baselines such as PointPillars [2], CenterPoint [1], and PillarNet [3], FG-PFE delivers substantial performance gains with small computational overhead.

I. INTRODUCTION

Ensuring the safety and stability of autonomous vehicles depends on their ability to perceive 3D spaces both accurately and in real-time. Recently, the advent of deep learning has spurred significant advancements in the field of LiDAR-based 3D object detection. While the majority of existing research has focused on improving accuracy, the need for real-time responsiveness cannot be overstated in the real-world scenarios. Therefore, there is a growing need for research that prioritises the development of computationally efficient, real-time LiDAR-based 3D object detectors.

To date, a variety of successful LiDAR-based 3D object detectors have been proposed, including VoxelNet [4], SECOND [5], PointPillar [2], PV-RCNN [6], and CenterPoint [1]. Given the irregular and sparse nature of LiDAR point clouds, various architectures have been proposed to model point clouds. These methods can be roughly categorized into point-based methods [7], [8], 3D voxel-based methods [5], [4], [9], and 2D pillar-based methods [2], [3], [10]. Point-based methods [7], [8] generate features directly from LiDAR points, modeling their spatial distribution in a hierarchical manner. However, these techniques require large computational overhead due to iterative point clustering step. Both voxel-based and pillar-based methods group LiDAR points using a predefined grid structure and encode the points within each grid, resulting in features organized in uniform grids. Voxel-based methods [5], [4], [9] employ a 3D regular voxel structure to cluster LiDAR points and separately encode the points for each voxel element. Then, 3D convolution layers are applied to further increase an abstraction level of the voxel features. However, voxel-based techniques need to encode the points in numerous voxels at the same time, and the 3D convolution itself is also computationally demanding. On the other hand, Pillar-based methods [2], [3], [10] organize LiDAR points into a 2D grid in the x-y plane that forms a series of pillars. These methods significantly reduce the number of grid elements and the resulting features can be efficiently processed by 2D convolutions. As a result, pillar-based methods offer enhanced efficiency, enabling real-time processing at speeds exceeding 10 FPS (frames per second). Given these advantages, such methods are now considered as a promising architectural choice for commercial applications.

While pillar-based methods are known for their computational efficiency, they often fail to outperform alternative approaches. This performance gap may be attributed to the

*These authors contributed equally to this work.
†Corresponding author.
[1]Department of Artificial Intelligence, Hanyang University, Seoul, 04763, Korea.
[2]Department of Electrical Engineering, Hanyang University, Seoul, 04763, Korea.
[3]Department of Electrical and Computer Engineering, College of Liberal Studies, Seoul National University, Seoul, 08826, Korea.
{konyulpark,yckim,jhkoh,bwpark}@spa.hanyang.ac.kr
junwchoi@snu.ac.kr

insufficient capability to capture spatial point distributions of the pillar encoding. To bridge this gap, several studies, including PillarNet [3], PillarNext [10], and HVNet [11], have focused on enhancing the effectiveness of pillar feature encoding without compromising its computational speed. Efforts to address these challenges have concentrated on optimizing two critical aspects of pillar-based methods: the generation of pillar features and their subsequent encoding through convolutional layers. PillarNet [3] and PillarNeXt [10] incorporated hierarchical deep pillar feature extraction and multi-scale fusion neck modules to improve the convolutional encoding. HVNet [11] enhanced pillar representation by integrating an attention mechanism into multi-scale pillars.

In this study, our objective is to improve the pillar encoding process in order to enhance the accuracy of 3D object detection. We hypothesize that the subpar performance of pillar-based methods compared to 3D voxel-based methods stems from not effectively capturing the fine-grained distributions of LiDAR points within each pillar. This is due to all points within each pillar being compressed into a single feature, failing to capture the dynamic point distribution across both spatial and temporal domains.

We propose a new pillar feature encoding method, referred to as *Fine-grained Pillar Feature Encoding* (FG-PFE). Our observation is that the reduced performance of pillar-based encoding relative to 3D voxel-based encoding originates from the absence of vertical grids when encoding the points in each pillar. Inspired by this observation, we devise *Spatio-Temporal Virtual* (STV) grids to obtain a fine-grained pillar representation. STV grids group points based on their vertical grid, temporal scanning grid, and horizontal displacement grid. Within each pillar, points are individually processed by three distinct modules: Vertical PFE (V-PFE), Temporal PFE (T-PFE), and Horizontal PFE (H-PFE). V-PFE divides each pillar using a vertical virtual grid and encodes the points within each virtual cluster separately. It then performs a weighted aggregation of the feature vectors from different vertical grids through a channel attention module. Although this grid structure may seem akin to that used in voxel-based methods, V-PFE generates a single feature vector for each pillar, thus retraining a computational advantage over voxel encoding. T-PFE organizes points in each pillar based on the LiDAR's scanning order. It encodes points from each scan order separately, incorporating the ability to model their temporal distribution. H-PFE introduces pillar grids with various horizontal grid offsets to capture multiple perspectives of Bird's Eye View (BEV) features. Finally, the proposed FG-PFE aggregates the features derived from these three encoding modules adaptively using an *Attentive Pillar Aggregation* (APA) module.

We evaluate the performance of FG-PFE on the widely used nuScenes benchmark [12]. Fig. 1 shows the performance versus complexity plot of FG-PFE in comparison with three pillar-based baseline methods, PointPillars [2], CenterPoint [1] and PillarNet [3]. FG-PFE offers significant performance gains over all baseline methods. In particular, FG-PFE achieves a 3.7% increase in the nuScenes detection score (NDS) over PointPillars and a 1.0% improvement over PillarNet, all with small additional computational increases.

The key contributions of the paper are summarized as follows.

- We present FG-PFE, a novel pillar feature encoding method that promises performance improvement for LiDAR-based 3D object detection.
- We devise a virtual grid structure termed STV grids that can capture the fine-grained distribution of LiDAR points within a pillar. STV grids provide added granularity to represent points across vertical, temporal, and horizontal dimensions.
- We introduce three distinct modules: V-PFE, T-PFE, and H-PFE. Each module generates pillar features using virtual grids in the vertical, temporal, and horizontal dimensions, respectively. These features are aggregated to produce the final pillar features used for subsequent 2D convolutional encoding.
- Our evaluation demonstrates that by integrating the proposed FG-PFE into pillar-based baselines, we achieve substantial performance enhancements with only a minor increase in computational overhead.

II. RELATED WORK

Numerous strategies have been developed for LiDAR-based 3D object detection, which generally fall into two main categories based on the type of input representation they use: point-based methods and grid-based methods.

Point-based techniques, such as PointRCNN [7] and 3DSSD [8], directly process raw point clouds without requiring any preliminary processing. PointRCNN [7] utilized PointNet++ [13] to first segment out the foreground 3D points and then refined the proposals using the segmented features. Meanwhile, 3DSSD [8] introduced an F-FPS method that complements the traditional D-FPS, incorporating a set abstraction operation that enhances both regression and classification tasks. Although point-based methods are known for their high spatial accuracy and the ability to flexibly aggregate features, their computational demands can be a limiting factor, especially in large-scale point cloud scenarios.

Grid-based approaches partition irregular point clouds using uniform grid and encode them individually for each grid element. Voxel-based methods, such as VoxelNet [4] and SECOND [5] employed 3D voxel structure to produce 4D voxel features, which were further encoded by 3D convolutional layers. Particularly, SECOND leveraged the inherent sparsity of voxel structures to reduce computational complexity. Subsequent advancements, including FSD [14] and VoxelNext [9], have significantly reduced computational demands by incorporating sparse features within the detection head. Despite these improvements, voxel-based methods still face challenges in real-time applications due to the need to process a large volume of voxels and the computational burden of 3D convolution operations. The pillar-based approach introduced in PointPillars [2] organized point clouds

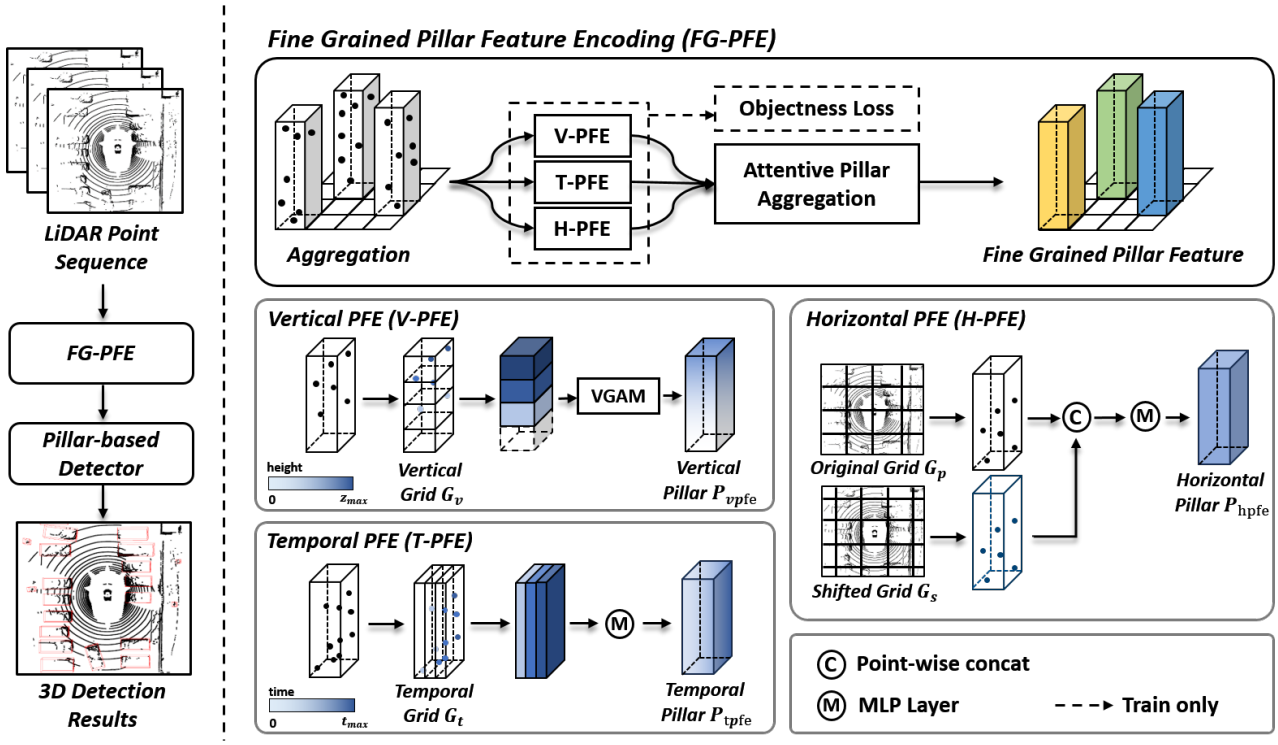

Fig. 2. **Overall architecture of the proposed FG-PFE.** LiDAR points are quantized along the vertical, temporal, and horizontal axes. In V-PFE, voxels from vertical axis are aggregated by the vertical grid attention module. In T-PFE, voxels from temporal axis are processed using a set of MLPs. In H-PFE, voxels from two different horizontal grid are transformed back into LiDAR points and then combined using concatenation. LiDAR points are then converted into pillar features with original pillar feature encoding. Pillar features originating from the three axes are combined using the Attentive Pillar Aggregation module.

into vertical columns called pillars. The features obtained from this pillar encoding were further processed through 2D convolution layers. Due to the absence of vertical grid and the use of 2D convolutions, this method significantly reduces computational load.

Recent research efforts have been directed at addressing the modeling limitations of PointPillars. PillarNet [3] and PillarNeXt [10] have enhanced the performance of PointPillars by integrating 2D sparse convolution layers into the backbone encoder and incorporating multi-scale fusion neck modules. Furthermore, HVNet [11] and Voxel-FPN [15] have enhanced performance by combining pillar features from various scales at the point-wise level. SST [16] and DSVT [17] used shifted pillar grid and rotated-set pillar grids to allow information exchange across different encoding schemes.

III. PROPOSED METHOD

In section, we present the details of the proposed FG-PFE.

A. Fine-Grained Pillar Feature Encoding

The overall architecture of FG-PFE is depicted in Fig. 2. Initially, LiDAR point clouds are structured into a conventional pillar grid as described in PointPillars [2]. Subsequently, the fine-grained distribution of points within each pillar is captured through simultaneous processing by V-PFE, T-PFE, and H-PFE. The features generated by these three encoding modules are then combined through the APA module. Consequently, the FG-PFE method generates pillar features that are of the same size as those produced by PointPillars. Finally, the consolidated pillar features are further processed by the subsequent stage of a typical 3D object detection framework.

1) Vertical PFE: V-PFE organizes points within each pillar by employing a virtual vertical grid, then encodes them to generate the pillar feature P_{vpfe}. Consider a scenario where the ith pillar contains a set of N_i LiDAR points, denoted as $p \in \mathbb{R}^{N_i \times 5}$. Each point is characterized by its (x, y, z) coordinates, reflectance value, and time lag. For brevity, we omit the pillar index i hereafter. As illustrated in Fig. 2, a vertical grid structure G_v with a size of H_p partitions the pillar and the points within each vertical grid element are encoded using mean-pooling. The features in pillars that are not empty (i.e., containing more than one non-empty vertical grid element) are gathered, generating the features P of dimensions $N_{vp} \times H_p \times C_p$, where N_{vp} denotes the count of non-empty pillars, and C_p denotes the channel dimension. This feature volume P is collapsed along a vertical dimension using the *Virtual Grid Aggregation Module* (VGAM). The overall VGAM process can be summarized as:

$$P' = M_c(P) \otimes P, \tag{1}$$

$$P'' = M_v(P') \otimes P', \tag{2}$$

$$P_{vpfe} = \psi(\text{reshape}(P'', N_{vp}, H_p \cdot C_p)), \tag{3}$$

where $\otimes$ denotes element-wise multiplication and $\psi(\cdot)$ denotes MLP layer. The channel attention operation $M_c(P)$ is expressed as

$$M_c(P) = \sigma(W_1(W_0(Avg(P)) + W_1(W_0(Max(P))), \quad (4)$$

where σ is the sigmoid function, $W_0 \in \mathbb{R}^{C/r \times C}$, and $W_1 \in \mathbb{R}^{C \times C/r}$, where r is the reduction ratio. Note that MLP weights W_0 and W_1 are shared for both inputs and the RELU function is followed by W_0. The process of vertical attention is expressed as

$$M_v(P) = \sigma(f(concat(Avg(P), Max(P)))), \quad (5)$$

where $f(\cdot)$ represents a 1D convolutional layer with the filter size of 7. Note that P'' is reshaped from the dimensions $N_{vp} \times H_p \times C_p$ to $N_{vp} \times (H_p \cdot C_p)$, and an MLP layer is employed to process the reshaped features and restore the original channel dimensions C_p, yielding the vertical pillar feature $P_{vpfe} \in \mathbb{R}^{N_{vp} \times C_p}$.

2) Temporal PFE: T-PFE models the temporal distribution of LiDAR inputs obtained from multiple sweeps. The LiDAR points within each pillar are organized using a temporal grid structure G_t of size T_p such that each grid element contains points from different sweeps. Similar to V-PFE, the points within each grid element are encoded using mean-pooling. The pillars containing more than one non-empty temporal grids are gathered, resulting in features of size $N_{tp} \times T_p \times C_p$, where N_{tp} denotes the number of non-empty temporal pillars and C_p denotes the channel dimension. The tensor P_{tp} is reshaped from $N_{tp} \times T_p \times C_p$ to $N_{tp} \times T_p C_p$ via grid-wise concatenation. Then, a MLP layer is used to transform back to its original channel dimension C_p, resulting in P_{tpfe}.

3) Horizontal PFE: The Horizontal PFE (H-PFE) utilizes multiple pillar grids to generate different pillar features. These grids are distinguished by different offsets in the horizontal domain. Consider two pillar grids shifted by half the grid size. Given a grid G_p, a shifted grid G_s is derived by translating G_p by half the grid size along the x and y axes. Subsequently, LiDAR points are partitioned using both the original and shifted grids, yielding the pillar features $P_{op} \in \mathbb{R}^{N_p \times C_p}$ and the shifted pillar features $P_{sp} \in \mathbb{R}^{N_p \times C_p}$. The pillar features obtained from the two distinct grids are concatenated in a point-wise manner. Finally, these combined features are arranged into the pillar grid G_p to generate the horizontal pillar features P_{hpfe}.

4) Attentive Pillar Aggregation: The APA combines three distinct features obtained from V-PFE, T-PFE, and H-PFE. The APA initially reduces the channel dimension to facilitate efficient feature fusion. Subsequently, it employs channel-wise attention (CWA) [18] in conjunction with convolutional layers to selectively aggregate the three pillar features. The outcome is final pillar features P_{fg} that maintain compatibility with existing pillar-based models. To summarize, the APA process unfolds as follows:

$$P_{fg} = \psi(CWA(concat(P_{vpfe}, P_{tpfe}, P_{hpfe}))), \quad (6)$$

where $CWA(\cdot)$ denotes the channel-wise attention module.

B. Objectness Score Prediction Loss

To enhance the individual features generated from the virtual grids, we introduce the concept of *objectness score prediction loss*, which compels the model to predict objectness scores for each virtual grid element. This loss encourages the model to discriminate between foreground and background grids, thus improving scene representation. The segmentation labels y_{seg} are directly derived from the 3D detection box annotations, indicating whether each grid is within or outside a ground-truth 3D box. The objectness score prediction is conducted by passing each pillar feature through a single MLP layer followed by a sigmoid function. The objectness score prediction loss function is given by

$$\mathcal{L}_{op} = \lambda_{gl}(l_{op}(P_{vpfe}) + l_{op}(P_{tpfe}) + l_{op}(P_{hpfe})), \quad (7)$$

where

$$l_{op}(P) = FL(\mathcal{A}(P)), y_{seg}), \quad (8)$$

and $FL(\cdot)$ denotes the focal loss function [27], $\mathcal{A}(\cdot)$ represents an MLP layer followed by a sigmoid function used for predicting objectness scores, and λ_{gl} is the balanced loss factor. The total loss for the entire network is the sum of the detection loss $\mathcal{L}_{det}$ and the objectness score prediction loss $\mathcal{L}_{op}$.

C. Modification on Group Head

The group head proposed in CBGS [19] organizes classes into hierarchical groups based on similar shapes and sizes. This allows the model distinguish objects within groups that share similar shape and size attributes. However, our experiments found that when we use separate group heads for bicycle and motorcycle classes, we observed a significant improvement in Average Precision (AP) over conventional grouping. This suggests that grouping classes of similar attributes like motorcycles and bicycles together does not necessarily lead to better performance. Our modified group head shows notable performance gains compared to conventional designs.

IV. EXPERIMENTS

A. nuScenes Dataset

The nuScenes dataset [12] constitutes a large-scale autonomous driving dataset comprising 1000 scenes, which consist of 700 training scenes, 150 validation scenes, and 150 test scenes. Each scene comprises LiDAR point cloud data acquired at a rate of $20Hz$ using a 32-channel LiDAR system. Annotated samples are provided at $2Hz$ and contain 1.4 million 3D bounding boxes. We follow official evaluation protocol for 3D object detection and evaluate mean Average Precision (mAP) and the nuScenes Detection score (NDS) across 10 foreground classes: barrier, bicycle, bus, car, motorcycle, pedestrian, trailer, truck, construction vehicle, and traffic cone.

TABLE I

QUANTITATIVE COMPARISON WITH STATE OF THE ART METHODS ON NUSCENES TEST SET. C.V AND T.C PRESENTS THE CONSTRUCTION VEHICLE AND THE TRAFFIC CONE, RESPECTIVELY, PED. AND MOTOR. ARE SHORT FOR THE PEDESTRIAN AND THE MOTORCYCLE, RESPECTIVELY. L INDICATES THE LIDAR. P/V/R DENOTES THE PILLAR, VOXEL AND RANGE VIEW BASED GRID ENCODER, RESPECTIVELY. THE BEST PERFORMANCE IS BOLDED

Method	Encoder	mAP	NDS	Latency	Car	Truck	Bus	Trailer	C.V	Ped.	Motor.	Bicycle	T.C	Barrier
CBGS [19]	V	52.8	63.3	-	81.1	48.5	54.9	42.9	10.5	80.1	51.5	22.3	70.9	65.7
CVCNet [20]	V+R	55.8	64.2	-	82.7	46.1	45.8	46.7	20.7	81.0	61.3	34.3	69.7	69.9
HotSpotNet [21]	V	59.3	66.6	-	83.1	50.9	56.4	53.3	23.0	81.3	63.5	36.6	73.0	71.6
CenterPoint [1]	V	58.0	65.5	106ms	84.6	51.0	60.2	53.2	17.5	83.4	53.7	28.7	76.7	70.9
Focals Conv [22]	V	63.8	70.0	-	86.7	56.3	67.7	59.5	23.8	87.5	64.5	36.3	81.4	74.1
AFDetV2 [23]	V	62.4	68.5	-	86.3	54.2	62.5	58.9	26.7	85.8	63.8	34.3	80.1	71.0
UTVR-L [24]	V	63.9	69.7		86.3	52.2	62.8	59.7	33.7	84.5	68.8	41.1	74.7	74.9
VISTA [25]	V+R	63.0	69.8	-	84.4	55.1	63.7	54.2	25.1	82.8	70.0	45.4	78.5	71.4
Transfusion-L [26]	V	65.5	70.2	-	86.2	56.7	66.3	58.8	28.2	86.1	68.3	44.2	82.0	78.2
PointPillars [2]	P	30.5	45.3	31ms	68.4	23.0	28.2	23.4	4.1	59.7	27.4	1.1	30.8	38.9
PillarNet-18 [3]	P	65.0	70.8	63ms	87.4	56.7	60.9	61.8	30.4	87.2	67.4	40.3	82.1	76.0
Ours	P	**65.7**	**71.8**	69ms	85.2	55.5	61.6	61.8	29.2	86.5	72.5	48.6	82.5	73.8

TABLE II

THE EFFECTS OF FG-PFE MODULES: STV, OBJECTNESS SCORE PREDICTION LOSS AND MODIFIED GROUP HEAD.

Method	Module			Performance	
	STV	Loss	Head	NDS (%)	Latency
PillarNet-18				64.12	63ms
Ours	✓			$65.13_{\uparrow 1.01}$	68ms
	✓	✓		$65.43_{\uparrow 1.31}$	68ms
	✓	✓	✓	$65.58_{\uparrow 1.46}$	69ms

TABLE III

THE EFFECTS OF PROPOSED V-PFE, T-PFE AND H-PFE IN STV.

Method	Module			Performance	
	V-PFE	T-PFE	H-PFE	NDS (%)	Latency
PillarNet-18				64.12	63ms
FG-PFE	✓			$64.50_{\uparrow 0.38}$	65ms
	✓	✓		$64.83_{\uparrow 0.71}$	66ms
	✓	✓	✓	$65.13_{\uparrow 1.01}$	68ms

B. Implementation Details

We integrate our FG-PFE module with PillarNet-18 [3], which is a state-of-the-art pillar-based detector. Each LiDAR point is represented as a 5-dimensional vector, $(x, y, z, r, \Delta t)$, where (x, y, z) denotes the 3D coordinate of the point, r indicates the point's intensity, and Δt represents the temporal displacement from the keyframe within the range $[0, 0.5)$. We consider a detection range spanning $[-54m, 54m]$ along the X and Y axes, and $[-5m, 3m]$ along the Z axis. The LiDAR pillar grid maintains a spatial resolution of $(0.075m, 0.075m)$, resulting in a pillar structure composed of 1440×1440 pillars.

We trained our model from scratch using a batch size of 16. For optimization, we adopted a one-cycle learning rate policy spanning 20 epochs, with a maximum learning rate set to 0.001. Furthermore, we employed a data augmentation strategy encompassing random flipping, rotation, scaling, and ground truth box sampling [5]. We set the balanced loss factor λ_{gl} to 1. In the inference stage, we optimize performance by employing batch normalization folding. During evaluation on the nuScenes test set, we utilized the double-flip test-time augmentation technique, following the method proposed in CenterPoint. [1]. The model's latency was measured on a single NVIDIA TITAN RTX GPU.

C. Performance on nuScenes Test Set

Table I presents a comparative analysis of our model against previous LiDAR-only 3D object detectors on the nuScenes test set. The performance metrics of other LiDAR-based methods are sourced from the nuScenes leaderboard. Our model demonstrates superior performance, achieving an mAP of 65.7% and an NDS of 71.8%, surpassing other LiDAR-based approaches. Notably, our model exhibits an improvement of 0.7% in mAP and 1.0% in NDS over the PillarNet-18 baseline, while introducing a negligible runtime increase of only 6 milliseconds. This result highlights a significant enhancement in performance, particularly in the detection of bicycles, motorcycles, traffic cones, and buses.

D. Ablation Studies on nuScenes Valid Set

In this section, we conduct several ablation studies on the nuScenes validation set. To expedite these experiments, we trained our models using only 25% of the training set, while utilizing the entire validation set for evaluation.

1) Contribution of Main Modules: Table II illustrates the impact of STV, objectness score prediction loss, and the modified group head on overall performance. Integrating spatio-temporal information in the pillar encoding stage through STV yields a notable 1.01% enhancement in NDS with a mere $5ms$ latency. Furthermore, the objectness score prediction loss contributes an additional 0.3% improvement in NDS without incurring extra computation. Lastly, the incorporation of the modified group head to discriminate finer details results in a 0.15% increase in NDS with a latency of $2ms$. The combination of all three components yields

TABLE IV
THE GAINS OF FG-PFE ON SEVERAL BASELINES.

Method	Performance		
	mAP (%)	NDS (%)	Latency
PillarNet-18	55.26	64.12	63ms
+ FG-PFE	$56.32_{\uparrow 1.06}$	$65.13_{\uparrow 1.01}$	68ms
CenterPoint-PointPillars	45.30	57.10	31ms
+ FG-PFE	$47.95_{\uparrow 2.65}$	$59.06_{\uparrow 1.96}$	36ms
PointPillars	39.47	53.51	29ms
+ FG-PFE	$43.59_{\uparrow 4.12}$	$57.21_{\uparrow 3.7}$	35ms

a cumulative performance gain of 1.46% in NDS over the PillarNet baseline. This underscores the significant enhancement in detection performance achieved by the proposed method while maintaining computational efficiency

2) Contribution of V-PFE, T-PFE and H-PFE: Table III shows the contribution of each pillar encoding module within the FG-PFE to the overall performance. The V-PFE module, which is designed to encode the point distribution in vertical dimension within each pillar, provides an increase of 0.38% in NDS. The integration of T-PFE with V-PFE improves the NDS by 0.71%. Finally, the integration of all three modules boils down to a total performance gain of 1.01% in NDS over the PillarNet baseline.

3) Performance Gain over Different Baselines: Table IV presents the 3D object detection performance when FG-PFE is integrated into several pillar-based 3D detection baselines. Integrating FG-PFE into the PillarNet-18 model yields improvements of 1.06% in mAP and 1.01% in NDS. Moreover, when applied to the pillar-based CenterPoint model, FG-PFE provides higher performance gains of 2.65% in mAP and 1.96% in NDS. The integration of FG-PFE with the PointPillars method demonstrates a notable improvement of 4.12% in mAP and 3.7% in NDS. The increase in latency due to FG-PFE is negligible, ranging from only $5ms$ to $6ms$, underscoring the efficiency of the method across various 3D detection methods. These results conclusively demonstrate that FG-PFE consistently delivers performance enhancements across a spectrum of baseline models.

V. CONCLUSIONS

In this paper, we introduced FG-PFE, a novel approach to pillar feature encoding designed to effectively capture the dynamic distribution of point clouds across spatial and temporal dimensions. Our method leverages STV grids to achieve a fine-grained pillar representation of point cloud distributions in vertical, temporal, and horizontal dimensions. V-PFE enhances the point cloud representation by segmenting each pillar further using a vertical virtual grid. T-PFE encodes points obtained from different scanning orders, capturing the temporal distribution of the point clouds. H-PFE explores multiple perspectives in representing the BEV scenes by employing different horizontal grid offsets. We introduced an objectness score prediction loss that aligns the three modules towards a unified detection objective. As a plug-and-play approach, FG-PFE can be readily integrated into existing pillar-based detectors. Our evaluation on the nuScenes dataset demonstrates that FG-PFE delivers significant improvements over several pillar-based baselines with only a negligible increase in computational complexity.

VI. ACKNOWLEDGMENT

This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2020R1A2C2012146).

REFERENCES

[1] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 784–11 793.

[2] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[3] C. M. Guangsheng Shi, Ruifeng Li, "Pillarnet: Real-time and high-performance pillar-based 3d object detection," *Proceedings of the European conference on computer vision (ECCV)*, 2022.

[4] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[5] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[6] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 529–10 538.

[7] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[8] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 040–11 048.

[9] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "Voxelnext: Fully sparse voxelnet for 3d object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 674–21 683.

[10] J. Li, C. Luo, and X. Yang, "Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 567–17 576.

[11] M. Ye, S. Xu, and T. Cao, "Hvnet: Hybrid voxel network for lidar based 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1631–1640.

[12] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural Information Processing Systems*, vol. 30, 2017.

[14] L. Fan, F. Wang, N. Wang, and Z.-X. ZHANG, "Fully sparse 3d object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 351–363, 2022.

[15] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-fpn: Multi-scale voxel feature aggregation for 3d object detection from lidar point clouds," *Sensors*, vol. 20, no. 3, p. 704, 2020.

[16] L. Fan, Z. Pang, T. Zhang, Y.-X. Wang, H. Zhao, F. Wang, N. Wang, and Z. Zhang, "Embracing single stride 3d object detector with sparse transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8458–8468.

[17] H. Wang, C. Shi, S. Shi, M. Lei, S. Wang, D. He, B. Schiele, and L. Wang, "Dsvt: Dynamic sparse voxel transformer with rotated sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 520–13 529.

[18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
[19] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.
[20] Q. Chen, L. Sun, E. Cheung, and A. L. Yuille, "Every view counts: Cross-view consistency in 3d object detection with hybrid-cylindrical-spherical voxelization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 224–21 235, 2020.
[21] Q. Chen, L. Sun, Z. Wang, K. Jia, and A. Yuille, "Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 2020, pp. 68–84.
[22] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5428–5437.
[23] Y. Hu, Z. Ding, R. Ge, W. Shao, L. Huang, K. Li, and Q. Liu, "Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 969–979.
[24] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia, "Unifying voxel-based representation with transformer for 3d object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 442–18 455, 2022.
[25] S. Deng, Z. Liang, L. Sun, and K. Jia, "Vista: Boosting 3d object detection via dual cross-view spatial attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8448–8457.
[26] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1090–1099.
[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.