2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11246382

Dense Semantic Bird-Eye-View Map Generation from Sparse LiDAR Point Clouds via Distribution-aware Feature Fusion

Jinsong Li, Kunyu Peng, and Yuxiang Sun

***Abstract*— Semantic scene understanding in bird-eye view (BEV) plays a crucial role in autonomous driving. A common approach to generating BEV maps from LiDAR point-cloud data involves constructing a pillar-level representation by projecting 3D point clouds onto a 2D plane. This process partially discards spatial geometric information, and produces sparse semantic maps. However, downstream tasks (e.g., trajectory planning and prediction), typically require dense grid-like semantic BEV maps rather than sparse segmentation outputs. To bridge this gap, we propose PointDenseBEV, an end-to-end, distribution-aware feature fusion framework. It takes as input sparse LiDAR point clouds and directly generates dense semantic BEV maps. Spatial geometric information and temporal context are embedded as auxiliary semantic cues within the BEV grid representation to enhance semantic density. Extensive experiments on the SemanticKITTI dataset demonstrate that our method achieves competitive performance compared to existing approaches.**

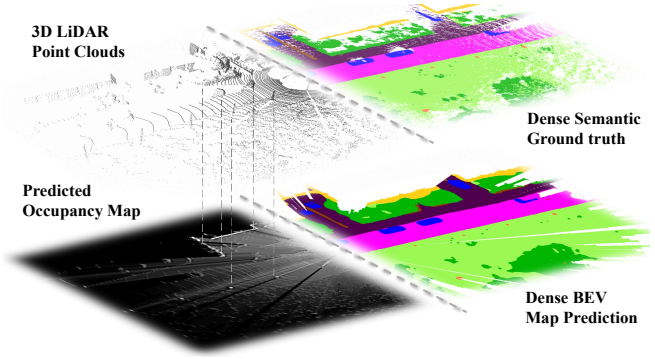


Fig. 1. Our method extracts distribution features from the raw LiDAR point clouds, which quantify the distribution variations of the point clouds along the vertical and horizontal directions (dashed lines show the correspondence between point clouds and 2D occupancy prediction). These spatial information is explicitly embedded into the BEV-grid features. The dense semantic BEV map are generated through learning ground-truth labels.

I. Introduction

Semantic scene understanding is a fundamental task for robots and autonomous vehicles [1-5]. A widely-adopted approach is semantic segmentation with visual images or LiDAR point clouds, which aims to assign a semantic label to each pixel or point. Compared to visual images, LiDAR point clouds [6,7] demonstrate robustness to varying lighting and weather conditions, making them an effective solution for semantic scene understanding. The processing of the unstructured LiDAR point-cloud data has shifted from traditional handcrafted methods [8] to representation learning [9] with the advancement of deep learning technologies [10] and the availability of large-scale datasets [11,12].

For autonomous driving, however, semantic scene understanding in bird-eye-view (BEV) [13-15] would be more attractive than directly segmenting pixels or points in raw data. This is because the downstream tasks in autonomous driving, such as trajectory planning or prediction, can be directly implemented with semantic BEV maps. To generate semantic BEV maps, we can simply project the 3D semantic point-cloud segmentation results into the 2D BEV plane. But this simple method leaves many holes in the BEV map (we call sparse BEV map in this paper), which is undesirable by the downstream tasks. Another method for generating semantic BEV maps is to split the point clouds into vertical pillars and label the points in each pillar into a semantic class [16]. However, this method still produce sparse BEV maps with holes.

One effective approach to mitigating the sparsity issue is to generate dense labels from multiple superimposed scans to refine the sparse BEV map prediction [17]. However, this method still operates within the pillar-based framework, in which the pillar operation directly compresses 3D data into 2D, resulting in the loss of geometric or semantic features. Some studies have attempted to alleviate the information loss during the pillar operation, including applying 3D convolution [18] and attention mechanisms to the point encoding process [19,20]. Other methods [21,22] have benefited from specialized convolutions or redesigned 2D backbones, enriching the semantic features.

Most of these pillar-based methods focus on aggregating the encoded pillar features to enhance the back-end feature representation, which is then fed into the task head. The inherent property, such as density or distribution, which reflects the richness of spatial information by quantifying point-cloud transmission and reflection in the 3D grid, has not been thoroughly explored. We believe that this inherent property not only compensates for the loss of geometric details caused by dimensionality reduction but also quantifies the contribution of local point features to pillar-level representations. Inspired by the research on 3D occupancy status of LiDAR point clouds [23], we explore how to embed this geometric spatial information into initial encoding process to complement and enhance grid features in the BEV map.

This work was supported in part by Hong Kong Research Grants Council under Grant 15222523, and in part by City University of Hong Kong under Grants 9610675. *(Corresponding author: Yuxiang Sun.)*

Jinsong Li and Yuxiang Sun are with the Department of Mechanical Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong (e-mail: jinsong.li@my.cityu.edu.hk; yx.sun@cityu.edu.hk, sun.yuxiang@outlook.com).

Kunyu Peng is with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany (e-mail: kunyu.peng@kit.edu).

In this paper, we propose a distribution-aware feature fusion method, named PointDenseBEV, to generate dense semantic BEV maps (not the sparse map with holes) from sparse LiDAR point clouds. An overview of the proposed method is illustrated in Fig. 1. Our method initiates with density quantification and occupancy estimation of point clouds in 3D grids. These distinguishable features are then incorporated as geometric supplements and contribution weights during 3D-to-2D projection, thereby mitigating the loss of spatial information. Furthermore, we perform temporal alignment of multi-frame data to increase the density of the semantic grid. Our code is open-sourced[1]. The main contributions of this paper are summarized as follows:

- We propose a multi-feature fusion framework that combines spatial and temporal feature aggregation to generate dense semantic BEV maps.
- We introduce a hierarchical pillar encoding module that utilizes across-voxel and voxel-level attention mechanisms to integrate LiDAR's spatial geometric features into pillar representations.
- We conduct extensive experiments on the SemanticKITTI dataset and discuss the effectiveness of each module.

II. Related Work

A. Point-cloud Processing

Existing approaches for point-cloud processing can be generally categorized into point-based and grid-based methods. Point-based methods [24] usually employ Multilayer Perceptrons (MLPs) and neighborhood query algorithms to encode and aggregate point features. While effectively preserving point clouds' spatial geometry for point-level representations, these methods are typically applied to small-scale scenarios. Grid-based methods discretize the 3D space into fixed-resolution voxels. An early work is VoxelNet [25], which learns a unified representation from grouped points within each voxel using a feature extractor (VFE). Recent works have made progress in refining the voxel-level features, including but not limited to the design of 3D convolution modules [26,27], multi-scale feature fusion modules [28,29] and non-uniform partitioning [30]. To mitigate the computational bottlenecks associated with 3D convolution, some works [16,31] partition point clouds into vertical pillars instead of voxel grids. This pillar-based paradigm enables direct 2D convolution processing while unifying the point cloud features into the BEV space. Related improvements [18,20,21] focus on feature fusion or designing efficient backbones to improve the performance of downstream tasks. However, the inherent height compression in pillar projection risks semantic ambiguity due to the loss of vertical distribution patterns during feature encoding.

B. LiDAR-based Semantic Scene Understanding

A fundamental task in scene understanding is to predict point-level or pixel-level (in the case of BEV grids) semantic labels for encoded point cloud features. For point-based methods, MLPs, Point Convolution [32], or Graph Nets [33] can be employed to establish feature-to-label mappings. For the discretized voxel-based methods, 3D convolutions can directly predict semantic labels from voxel features. Choy et al. [34] first designed a sparse voxel framework that reduces unnecessary calculations for empty volumes. Thus, sparse 3D convolution has gradually become the primary choice for voxel processing [35,36], although it still needs a certain computation cost. For projection-based methods, 2D convolutions can be used for semantic prediction on spherical surfaces [37], or top view [38]. However, projection transformations inherently retain point-cloud sparsity, causing the predicted semantics to be fragmented. To mitigate this, Bieder et al. [39] stacked semantic labels from multiple scans to train a mapping from sparse point clouds to dense semantic grids. Subsequent works [17,19] incorporate visibility maps to infer semantics for empty pillars, establishing a comprehensive framework for this task.

As discussed previously, 3D-to-2D projection via pillar partitioning discards the original geometric information of the point cloud. In contrast to methods that focus on back-end network design, our work concentrates on the front-end by explicitly embedding spatial density information during feature encoding, thereby enhancing geometric awareness and semantic discriminability of the BEV map.

III. The Proposed Method

A. The Method Overview

As shown in Fig. 2, our PointDenseBEV takes as input LiDAR point clouds from multiple temporal frames. We first explore the extraction of point-cloud spatial features through the transmission properties, and then discuss how to leverage this information, especially along the vertical dimension, to enhance feature representations at different levels.

B. Feature Encoding and Aggregation

1) Spatial Feature Extraction: LiDAR point clouds typically exhibit a layered or banded distribution in 3D space. From this sparse and uneven distribution, we can infer two important spatial properties: (1) vertical density variation, and (2) grid occupancy estimation.

Given a point cloud, the 3D space is first voxelized into a spherical grid with resolution $(\Delta\rho, \Delta\theta, \Delta\phi)$, where ρ, θ, and ϕ respectively denote radial distance, azimuthal angle, and polar angle. Each voxel carries two attributes:

$$R(\rho,\theta,\phi) = \sum_{i\in N} \frac{V_i}{V_{\text{voxel}}}, \quad T(\rho,\theta,\phi) = \sum_{\rho'>\rho} R(\rho',\theta,\phi), \quad (1)$$

where the reflection $R(\rho,\theta,\phi)$ represents the weighted sum of contributions from N neighboring points, computed by the spatial overlap between V_i point's unit space V_i and the current voxel. The transmission $T(\rho,\theta,\phi)$ accumulates reflections along the radial direction, from the farthest voxel toward the LiDAR sensor. These attributes are then mapped into Cartesian voxel grid with resolution $(\Delta x, \Delta y, \Delta z)$ to form a density map $\mathcal{M}^{\text{tr}} \in \mathbb{R}^{W\times H\times N_v\times 2}$, where $W \times H$

[1] https://github.com/lab-sun/PointDenseBEV

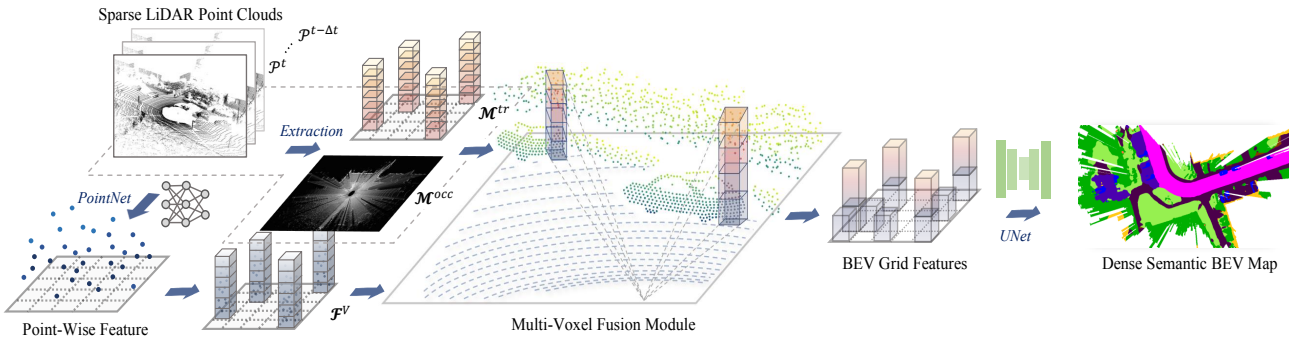


Fig. 2. The overall architecture of our PointDenseBEV. Spatial geometric information – the occupancy map $\mathcal{M}^{occ}$ and density map $\mathcal{M}^{tr}$, is extracted from raw point clouds. These features, together with voxel-wise features, are fed into the Multi-Voxel Fusion Module to generate BEV grid features. The BEV grid features are processed by a modified UNet architecture, of which the output is then mapped to a dense BEV map using two 1×1 convolutions.

denotes the grid dimensions and N_v is the vertical voxel count.

We assume that each voxel can be in one of two states $\Omega = \{Occ, Free\}$ representing either occupied or free state. Probabilities are assigned via a mapping $f : 2^{\Omega} \rightarrow [0, 1]$. In Cartesian coordinates, we adopt the same probabilistic hypothesis model proposed by Kalble et al. [23]:

$$f(\omega|T,R) = \begin{cases} p_{\text{FN}}^{T}(1-p_{\text{FP}}^{R}) & \text{for } \omega = Occ, \\ p_{\text{FP}}^{R}(1-p_{\text{FN}}^{T}) & \text{for } \omega = Free, \\ 1 - m(Occ) - m(Free) & \text{for } \omega = \Omega. \end{cases} \tag{2}$$

We adopt the false negative/positive probabilities $p_{\text{FN}} = 0.9$ and $p_{\text{FP}} = 0.1$. The occupancy map $\mathcal{M}^{\text{occ}} \in \mathbb{R}^{W \times H}$ is constructed by aggregating binary values along the Z-axis:

$$\begin{aligned} s(x,y) &= \frac{1}{n_v} \sum_{z} s(x,y,z), \\ s(x,y,z) &= \begin{cases} 1 & \text{if } f(Occ) > f(Free), \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{3}$$

n_v is the number of non-empty voxels in vertical space at each fixed (x,y) location.

2) Voxel-Level Feature Fusion: Pillar-based methods generate 2D pseudo-images by selectively sampling point features for efficient 2D convolution, but lose geometric information during height compression. To compensate for this compromise in feature encoding, we propose a novel feature fusion module to restore lost spatial priors through hierarchical integration of $\mathcal{M}^{\text{tr}}$ and $\mathcal{M}^{\text{occ}}$.

In the $W \times H \times N_v$ voxel grid, each non-empty voxel is assigned 10 points, obtained through sampling or padding. Each point is augmented by calculating coordinate offsets, yielding a 9D vectors $\mathcal{P} = \{p_i \in \mathbb{R}^9\}_{i=1}^{N_p}$, N_p is the total point count. Here, we use a simplified PointNet (SPN) to encode $\mathcal{P}$ into point-wise features, followed by a max-pooling operation to obtain voxel representations with C_v channels:

$$\mathcal{F}^{V} = Max\left(\text{SPN}\left(\mathcal{P}\right)\right), \mathcal{F}^{V} \in \mathbb{R}^{N_v \times C_v}. \tag{4}$$

The Multi-Voxel Fusion (MVF) module compresses voxel information along the vertical dimension using the density map $\mathcal{M}^{tr}$, generating pillar features enriched with spatial distribution. As Fig. 3 illustrates, MVF module contains two modified attention mechanisms. The density map $\mathcal{M}^{\text{tr}}$ is first processed through a two-layer MLP with C_m output channels and then reshaped into $\mathcal{F}^{M} \in \mathbb{R}^{WH \times N_v \times C_m}$. Subsequently, the query Q, key K, and value V matrices are obtained by projecting the features with weight matrices:

$$Q = \mathcal{F}^{V} W^{Q}, \quad K = \mathcal{F}^{M} W^{K}, \quad V = \mathcal{F}^{M} W^{V} \tag{5}$$

We use the dot-product operation to construct the correlation matrix between the point-cloud spatial distribution and voxel features in the same vertical volume, followed by Softmax normalization:

$$\mathcal{F}' = \text{Softmax}\left(\frac{QK^{\top}}{\sqrt{d}}\right) V + \mathcal{F}^{V}, \tag{6}$$

where $\mathcal{F}' \in \mathbb{R}^{WH \times N_v \times C_p}$ is the intermediate feature vector.

In the second stage, voxel-level attention is applied to compress the feature map by modeling inter-voxel relationships along the vertical axis:

$$\mathcal{F}'' = M_c(\mathcal{F}', \mathcal{F}^{M}) \otimes \mathcal{F}', \tag{7}$$

where $\otimes$ is element-wise multiplication. The voxel attention M_c is computed as:

$$\begin{aligned} M_c(\mathcal{F}', \mathcal{F}^{M}) =& \sigma(W_1(W_0(Avg(\mathcal{F}'))) + W_1(W_0(Max(\mathcal{F}'))) \\ &+ W_1{}'(W_0{}'(Sum(\mathcal{F}^{M})))), \end{aligned} \tag{8}$$

where W_1, W_0, W_1' and W_0' denote the weight of two-layers MLPs. In Eq. 8, the summed feature $\mathcal{F}^{M}$ provides supplementary spatial cues to guide the attention mechanism toward regions of higher point density. After sigmoid activation σ, $\mathcal{F}''$ is reshaped to $W \times H \times N_v C_p$ and passed through an MLP to obtain the final pillar features:

$$\mathcal{F}^{P} = \text{MLP}(\text{reshape}(\sigma(\mathcal{F}'')) \in \mathbb{R}^{W \times H \times C_p} \tag{9}$$

Finally, an occupancy network composed of two 3×3 Conv+Norm+ReLU blocks is used to encode the occupancy map $\mathcal{M}^{occ}$. The outputs is then concatenated with $\mathcal{F}^{P}$, forming the distribution-aware pillar representations on the X-Y grid.

3) Scene-level Frame Aggregation: Numerous studies have shown that aggregating multi-frame temporal information significantly enhances 3D detection performance. We

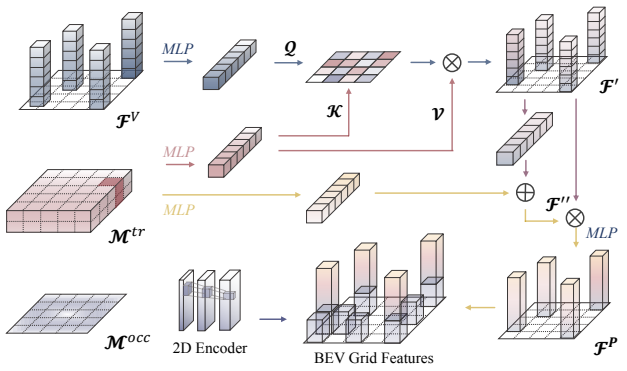


Fig. 3. The Illustration of our proposed Multi-Voxel Fusion (MVF) module. Occupancy map $\mathcal{M}^{occ}$ and voxel feature $\mathcal{F}^V$ are sent to module to compute $\mathcal{Q}, \mathcal{K}, \mathcal{V}$ and cross-voxel correlation.

believe that increasing the spatial density of points improves feature correspondences between pillar and grid cells in the BEV space. To this end, points from historical LiDAR scans are aggregated into the current scene through coordinate transformation.

Let $\mathcal{P}^t$ denote the point cloud at time step t. The merging of historical point cloud frames over Δt is formulated as:

$$\mathcal{P}^t_{\text{merged}} = \bigcup_{i=t-\Delta t}^{t} T_{i\to t}\mathcal{P}^i, \tag{10}$$

where $T_{i\to t}$ is the transformation matrix aligning the historical point cloud $\mathcal{P}^i$ to the current time step t, given by:

$$T_{(t-\Delta t)\to t} = T^{-1}_{t\to\text{world}} \cdot T_{(t-\Delta t)\to\text{world}}. \tag{11}$$

The aggregated point cloud is then fed into the network for pillar feature extraction. However, this process inevitably increases computation and memory costs, and incorporating more historical frames does not always lead to better performance. In the experimental section, we analyze the impact of the time window size Δt.

C. Network Architecture

The encoder-decoder [40] architecture has demonstrated strong performance and generalizability across domains like medical imaging and autonomous driving. In this work, we adopt a modified U-Net as the 2D backbone, integrated with the feature encoding network and the task head to construct a complete pipeline for dense BEV map prediction.

1) Backbone Design: The backbone receives input tensors of shape $B \times W \times H \times C$, where $C = 128$ is the sum of the pillar feature channels and the encoded channels of $\mathcal{M}^{\text{occ}}$. Both the downsampling and upsampling paths are reduced to four layers (originally five). In addition, each layer incorporates residual connections within each 3×3 convolution block, while the rest of the architecture remains unchanged. A two-layer output head is appended to the end of the 2D backbone, implemented with two consecutive 1×1 convolutional layers, which jointly perform feature transformation and semantic class mapping.

2) Loss Function: Following the training setup of MASS [19], we adopt a weighted cross-entropy loss for the BEV semantic segmentation task, which is defined as:

$$\mathcal{L}_{\text{wce}}(y, \hat{y}) = -\sum \omega_i p(y_i) \log(p(\hat{y}_i)),$$

where $p(y_i)$ is the ground-truth label for pixel i, $p(\hat{y}_i)$ is the predicted label, and ω_i is the class-specific weight. In the 2D BEV map, only pixels with valid ground-truth labels contribute to the gradient during training.

IV. Experimental Results and Discussions

A. Datasets and Evaluation Metrics

Our proposed PointDenseBEV was trained and evaluated on the SemanticKITTI dataset [11]. For fair comparison with model [17], we adopt the same data splitting: training on sequences 00-07 and 09-10, and validating on sequence 08. The same dense/sparse label conventions and prediction class definitions are maintained. For more details, refer to the related works [19].

The performance is evaluated using the Intersection-over-Union (IoU) metric. For each class i, the IoU_i is computed as $\text{IoU}_i = \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i+\text{FN}_i}$, where TP_i, FP_i, and FN_i denote true positives, false positives, and false negatives, respectively. Higher IoU_i (closer to 1) indicates better prediction performance. The overall model performance is quantified by the mean IoU (mIoU) across all N classes: $mIoU = \frac{1}{N}\sum_{i=1}^{N} \text{IoU}_i$.

B. Implementation Details

1) Model: Each LiDAR point is represented as a vector of (x, y, z) coordinates and reflectivity r. To define the valid 3D space, the point range is constrained to a Cartesian grid with limits of $[-50.0m, 50.0m]$ in the X-axis, $[-25.0m, 25.0m]$ in the Y-axis, and $[-2.5m, 1.5m]$ in the Z-axis. For reflectance and refraction computations, the spherical grid is cropped based on r, θ, with respective ranges of $[2.5m, 60m]$, $[5/12\pi, 25/36\pi]$, and $[-\pi, \pi]$. The maximum number of points per pillar is set to $N_p = 20$, and per voxel to $N_v = 10$, with 50,000 voxels in the 3D grid.

2) Training: The training pipeline is built upon the open-source project OpenPCDet [41] with same data augmentation strategies from [17]. All experiments are conducted on an NVIDIA RTX A6000 GPU. We use the Adam optimizer with a cosine annealing learning rate schedule in 20 epochs. The learning rate η linearly increases to 0.003 over the first 10% of the training epochs and then decays to $10e-5$ for the remaining epochs. The weight decay is set to 0.01, momentum to 0.9, and the batch size to 2.

C. Quantitative Results

To evaluate the effectiveness of our PointDenseBEV, three existing dense semantic prediction approaches: Bieder et al [39], PillarSegNet [17], and MASS [19] are used as baselines. We list the experimental results of the first two methods in quantitative analysis. In the original MASS implementation, a raycasting-based visibility map was introduced as an additional input stream to the network. Here, we replace this input with our occupancy map described in the previous section and provide the reproduced results.

TABLE I

COMPARATIVE RESULTS IN TERMS OF IOU (%) ON THE VALIDATION SET OF SEMANTICKITTI. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Setting	Methods	mIoU	Vehicle	Person	Two-wheel	Rider	Road	Side-walk	Other-ground	Building	Object	Vege.	Trunk	Terrain
Sparse Mode	Bieder et al.	39.8	69.7	0.0	0.0	0.0	85.8	60.3	25.9	72.8	15.1	68.9	9.90	69.3
	Pillar	55.1	79.5	15.8	25.8	51.8	89.5	70.0	38.9	80.6	25.5	72.8	38.1	72.7
	Pillar + Vis	55.3	82.7	20.3	24.5	51.3	90.0	71.2	36.5	81.3	28.3	70.4	38.5	69.0
	MASS	58.8	**85.8**	34.2	26.8	58.5	**91.3**	74.0	38.1	82.2	28.7	79.5	35.7	71.3
	MASS + Occ	59.1	83.8	33.1	23.5	60.6	89.6	71.8	32.7	**84.6**	33.8	**83.3**	35.5	**77.6**
	Ours	**61.9**	85.0	**37.6**	**30.0**	**64.5**	90.5	**74.4**	**42.5**	83.8	**33.9**	82.4	**42.7**	75.3
Dense Mode	Pillar	42.8	70.3	5.4	6.0	8.0	89.8	65.7	34.0	65.9	16.3	61.2	23.5	67.9
	Pillar + Vis	44.1	72.8	7.4	4.7	10.2	90.1	66.2	32.4	67.8	17.4	63.1	27.6	69.2
	MASS	44.9	72.1	6.8	6.2	9.9	90.1	65.8	**37.8**	67.1	18.8	68.1	24.7	71.4
	MASS + Occ	45.1	69.9	9.1	5.2	**17.3**	89.7	65.1	27.9	71.5	**21.8**	**70.7**	20.7	**73.7**
	Ours	**46.4**	69.2	**16.0**	**17.3**	10.4	89.4	66.3	33.0	70.4	18.1	68.7	25.8	72.3
	Ours + His	**47.0**	**73.9**	11.8	11.4	13.8	**90.6**	**68.0**	34.4	**71.6**	20.2	67.3	**28.9**	69.8

1) Comparative Studies: We evaluate all methods under two distinct settings: one generates sparse semantic maps mimicking the input point cloud's distribution, and the other one produces dense semantic maps through label accumulation. Tab. I summarizes the performance of all evaluated methods. In the reproduction of the MASS, incorporating the occupancy map (MASS + Occ) achieve improvement of 0.3% and 0.2% percentage in the sparse and dense modes respectively. The PointDenseBEV achieves 61.9% mIoU in the sparse mode, outperforming all the baselines.

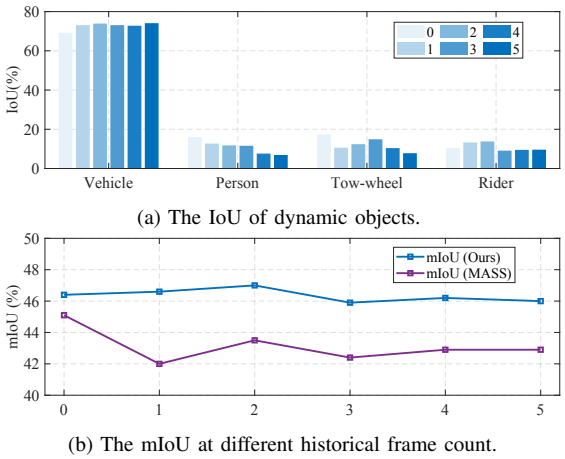
Additionally, Tab. I reports two representative results of PointDenseBEV in the dense setting. Without multi-frame aggregation, it outperforms the best model MASS by 1.2%. When incorporating two previous frames, performance further increases to 47.0% mIoU. Notably, PointDenseBEV excels in processing dynamic classes (Vehicle, Person, etc.) occupying limited vertical space, which is also crucial for downstream dynamic navigation planning. This phenomenon may benefit from pillar encoding's ability to capture important regions. In the MVF module, the spatial distribution of point clouds is not only integrated into voxel features but also used to evaluate voxel-wise importance during fusion. For ground classes (Road, Sidewalk, etc.), the performance differences across methods are relatively minor.

TABLE II

THE EFFECTS OF HISTORICAL FRAMES IN TERMS OF MIOU (%).

Method	History frame	mIoU	Latency
Pillar	0	42.8	-
MASS	0	44.9	-
Ours	0	46.4	41 ms
	1	46.6	44 ms
	2	47.0	47 ms
	3	45.9	51 ms
	4	46.2	53 ms
	5	46.0	54 ms

2) The Effects of Historical Frames: To further validate the aggregating temporal information, we compare mIoU scores for different numbers of aggregated frames under a fixed voxel count (50,000). Tab. II presents the comparison of accuracy and efficiency, where the number of input historical frames is varied from 0 to 5. Our method consistently outperforms the baseline in mIoU, though its latency grows with frame count. As shown in the results, aggregating more frames does not monotonically improve semantic prediction performance. Optimal performance is achieved with 2 historical frames, after which the mIoU begins to decline.



(a) The IoU of dynamic objects.

(b) The mIoU at different historical frame count.

Fig. 4. The effect of temporal information. We compared the prediction results of dynamic objects under different numbers of historical frames (a), and the mIoU of the two methods (b).

Fig. 4(a) illustrates the IoU scores on dynamic classes (Vehicle, Person, Two-wheel, Rider) with different historical frame inputs. The decline in mIoU may be attributed to the ghosting effect introduced by coordinate transformations when aligning multiple frames of moving objects, leading to semantic ambiguity in the corresponding BEV grid. Fig. 4(b) further compares our PointDenseBEV with MASS+Occ. As the number of input frames increases, MASS's performance does not improve. This may be due to its use of point-cloud level LSTM attention, where an increasing number of points

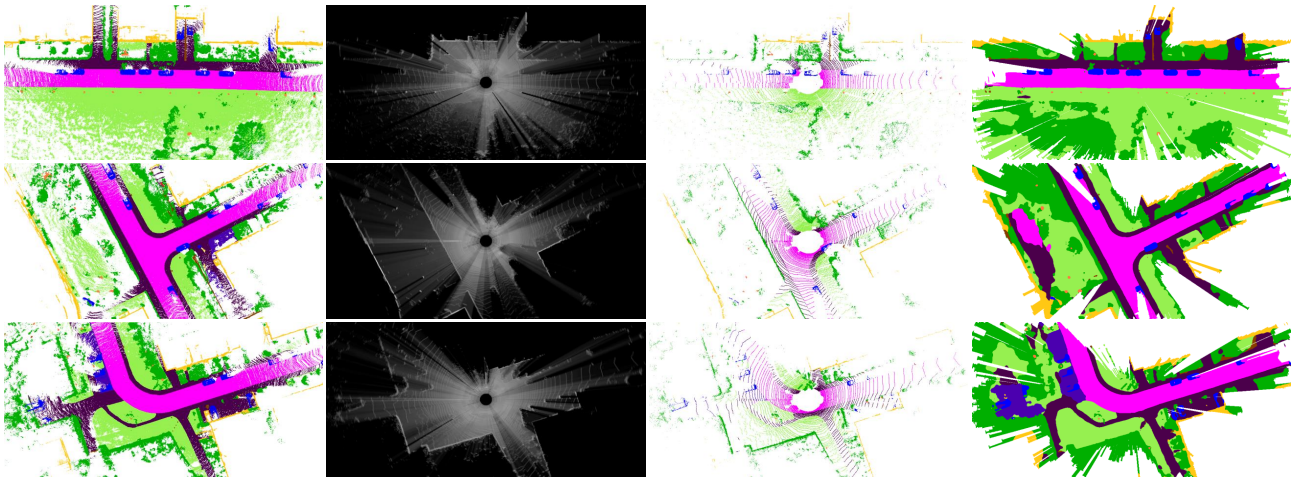


Fig. 5. Sample qualitative demonstrations for our BEV map generation on the SemanticKITTI dataset. We select three validation scenes from the dataset, presented from left to right as follows: dense semantic ground truth, predicted occupancy maps, generated sparse BEV map and dense BEV map.

TABLE III

THE EFFECT OF VOXEL PARTITIONING. Δ IS THE SCORE GAP BETWEEN THE CORRESPONDING PARTITION.

the number of voxels	mIoU(%)	Δ(%)
30,000	45.3	− 1.6
40,000	46.0	− 0.4
50,000	47.0	0.0
60,000	46.9	− 0.1
70,000	46.7	− 0.3

TABLE IV

ABLATION STUDY IN TERMS OF MIOU (%) FOR FEATURE ENCODING, THE BASELINE REFERS TO A METHOD THAT EXCLUDES OCC-MAP, HIS-FRAME INPUTS, MVF MODULE.

Method	Module			Performance	
	Occ-Map	His-Frame	MVF	mIoU	Latency
Baseline				42.57	39 ms
Ours	✓			45.34	39 ms
	✓		✓	46.42	41 ms
	✓	✓		46.01	42 ms
	✓	✓	✓	46.96	47 ms

negatively affects the module's performance.

3) The Effects of Voxel Number: Stacking consecutive frames introduces temporal information while also increases the number of point clouds in the current scene. This subsection conducts an analysis to verify the effect of voxel partitioning, using the aggregation of two historical frames as a case study. Tab. III presents the semantic prediction results over different voxel settings. A voxel number of 50,000 achieves the optimal balance between segmentation performance (i.e., mIoU) and computational efficiency. Fewer voxels lead to insufficient spatial detail, while too many can cause information redundancy.

We also perform ablation studies to evaluate the different proposed components. All experiments are conducted on the dense BEV semantic prediction task, and results are reported in Tab. IV. The results show that incorporating the occupancy map alone yields a significant improvement of 2.8% in mIoU over the baseline. Adding temporal and spatial distribution features further improves mIoU by 0.67% and 1.08%, respectively, with the latter showing a more pronounced effect. Combining all three strategies yields the best overall performance, which is the optimal model.

D. Qualitative Demonstrations

Fig. 5 presents sample qualitative demonstrations of our PointDenseBEV. In sparse BEV semantic prediction setting, PointDenseBEV achieves an result similar to semantic segmentation. The sparse BEV map retains precise boundaries for vehicles, while the dense BEV map fills in missing semantics and provided the complete shape, generating more continuous and coherent scene representations. For static structures such as roads and sidewalks, PointDenseBEV identifies coverage even in regions with extremely sparse LiDAR point clouds.

V. CONCLUSIONS AND FUTURE WORK

We presented PointDenseBEV, a distribution-aware feature fusion network designed to generate dense semantic BEV maps from sparse LiDAR point clouds. Unlike traditional pillar-based approaches that suffer from sparsity and information loss during 3D-to-2D projection, our method incorporates spatial distribution features and temporal aggregation to enhance BEV grid features. Specifically, our PointDenseBEV incorporates density and the predicted occupancy map as geometric supplementary features into the pillar encoding process through the MVF module. We validated the effectiveness of our PointDenseBEV in dense semantic prediction on the SemanticKITTI dataset and demonstrated the performance improvements contributed by each designed module. In the future, we would like to employ generative models to improve the performance.

REFERENCES

[1] Y. Wang, H. K. Chu, and Y. Sun, "Peafusion: Parameter-efficient adaptation for rgb-thermal fusion-based semantic segmentation," *Information Fusion*, p. 103030, 2025.

[2] J. Huang, J. Li, N. Jia, Y. Sun, C. Liu, Q. Chen, and R. Fan, "Roadformer+: Delivering rgb-x scene parsing through scale-aware information decoupling and advanced heterogeneous feature fusion," *IEEE Transactions on Intelligent Vehicles*, pp. 1–10, 2024.

[3] H. Li, H. K. Chu, and Y. Sun, "Temporal consistency for rgb-thermal data-based semantic scene understanding," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9757–9764, 2024.

[4] Y. Feng, W. Hua, and Y. Sun, "Nle-dm: Natural-language explanations for decision making of autonomous driving based on semantic scene understanding," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 9780–9791, 2023.

[5] W. Ma, S. Huang, and Y. Sun, "Triplet-graph: Global metric localization based on semantic triplet graph for autonomous vehicles," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3155–3162, 2024.

[6] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for lidar point clouds in autonomous driving: A review," *IEEE Transactions on Neural Networks and Learning System*, vol. 32, no. 8, pp. 3412–3432, 2020.

[7] H. Xu, H. Liu, S. Huang, and Y. Sun, "C2l-pr: Cross-modal camera-to-lidar place recognition via modality alignment and orientation voting," *IEEE Transactions on Intelligent Vehicles*, pp. 1–17, 2024.

[8] X.-F. Han, S.-J. Sun, X.-Y. Song, and G.-Q. Xiao, "3d point cloud descriptors in hand-crafted and deep learning age: State-of-the-art," *arXiv preprint arXiv:1802.02297*, 2018.

[9] L. Wang and Y. Huang, "A survey of 3d point cloud and deep learning-based approaches for scene understanding in autonomous driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 135–154, 2021.

[10] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE access*, vol. 7, pp. 53 040–53 065, 2019.

[11] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9297–9307.

[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 Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[13] Y. Feng and Y. Sun, "Polarpoint-bev: Bird-eye-view perception in polar points for explainable end-to-end autonomous driving," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2024.

[14] H. Li, C. Sima, J. Dai, W. Wang, L. Lu, H. Wang, J. Zeng, Z. Li, J. Yang, H. Deng *et al.*, "Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 4, pp. 2151–2170, 2023.

[15] S. Gao, Q. Wang, and Y. Sun, "S2g2: Semi-supervised semantic bird-eye-view grid-map generation using a monocular camera for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 974–11 981, 2022.

[16] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[17] J. Fei, K. Peng, P. Heidenreich, F. Bieder, and C. Stiller, "Pillarsegnet: Pillar-based semantic grid map estimation using sparse lidar data," in *Proceedings IEEE Intelligent Vehicles Symposium*. IEEE, 2021, pp. 838–844.

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

[19] K. Peng, J. Fei, K. Yang, A. Roitberg, J. Zhang, F. Bieder, P. Heidenreich, C. Stiller, and R. Stiefelhagen, "Mass: Multi-attentional semantic segmentation of lidar data for dense top-view understanding," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 824–15 840, 2022.

[20] D. T. Le, H. Shi, H. Rezatofighi, and J. Cai, "Accurate and real-time 3d pedestrian detection using an efficient attentive pillar network," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1159–1166, 2022.

[21] S. Zhou, Z. Tian, X. Chu, X. Zhang, B. Zhang, X. Lu, C. Feng, Z. Jie, P. Y. Chiang, and L. Ma, "Fastpillars: a deployment-friendly pillar-based 3d detector," *arXiv preprint arXiv:2302.02367*, 2023.

[22] W. Xu, S. Zhou, and Z. Yuan, "Pillartrack: Redesigning pillar-based transformer network for single object tracking on point clouds," *arXiv preprint arXiv:2404.07495*, 2024.

[23] J. Kälble, S. Wirges, M. Tatarchenko, and E. Ilg, "Accurate training data for occupancy map prediction in automated driving using evidence theory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5281–5290.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[25] 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.

[26] Y. Li, L. Fan, Y. Liu, Z. Huang, Y. Chen, N. Wang, and Z. Zhang, "Fully sparse fusion for 3d object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[27] S. Dong, L. Ding, H. Wang, T. Xu, X. Xu, J. Wang, Z. Bian, Y. Wang, and J. Li, "Mssvt: Mixed-scale sparse voxel transformer for 3d object detection on point clouds," *Advances in Neural Information Processing Systems*, vol. 35, pp. 11 615–11 628, 2022.

[28] S. Liu, W. Huang, Y. Cao, D. Li, and S. Chen, "Sms-net: Sparse multi-scale voxel feature aggregation network for lidar-based 3d object detection," *Neurocomputing*, vol. 501, pp. 555–565, 2022.

[29] M. Ye, S. Xu, and T. Cao, "Hvnet: Hybrid voxel network for lidar based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1631–1640.

[30] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9939–9948.

[31] G. Shi, R. Li, and C. Ma, "Pillarnet: Real-time and high-performance pillar-based 3d object detection," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 35–52.

[32] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6411–6420.

[33] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 296–10 305.

[34] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

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

[36] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 685–702.

[37] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2019, pp. 4213–4220.

[38] X. Li, G. Zhang, H. Pan, and Z. Wang, "Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2022, pp. 11 117–11 123.

[39] F. Bieder, S. Wirges, J. Janosovits, S. Richter, Z. Wang, and C. Stiller, "Exploiting multi-layer grid maps for surround-view semantic segmentation of sparse lidar data," in *Proceedings IEEE Intelligent Vehicles Symposium*. IEEE, 2020, pp. 1892–1898.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[41] O. D. Team, "Openpcdet: An open-source toolbox for 3d object detection from point clouds," https://github.com/open-mmlab/OpenPCDet, 2020.