Semantic-MoSeg: Semantics-Assisted Moving-Obstacle Segmentation in Bird-Eye-View for Autonomous Driving

Shiyu Meng, *Student Member, IEEE*, and Yuxiang Sun, *Member, IEEE*

***Abstract*— Bird-eye-view (BEV) perception for autonomous driving has become popular in recent years. Among various BEV perception tasks, moving-obstacle segmentation is very important, since it can provide necessary information for downstream tasks, such as motion planning and decision making, in dynamic traffic environments. Many existing methods segment moving obstacles with LiDAR point clouds. The point-wise segmentation results can be easily projected into BEV since point clouds are 3-D data. However, these methods could not produce dense 2-D BEV segmentation maps, because LiDAR point clouds are usually sparse. Moreover, 3-D LiDARs are still expensive to vehicles. To provide a solution to these issues, this paper proposes a semantics-assisted moving-obstacle segmentation network using only low-cost visual cameras to produce segmentation results in dense 2-D BEV maps. Our network takes as input visual images from six surrounding cameras as well as the corresponding semantic segmentation maps at the current and previous moments, and directly outputs the BEV map for the current moment. We also propose a movable-obstacle segmentation auxiliary task to provide semantic information to further benefit moving-obstacle segmentation. Extensive experimental results on the public nuScenes and Lyft datasets demonstrate the effectiveness and superiority of our network.**

***Index Terms*— Moving-obstacle segmentation, bird-eye-view perception, semantic segmentation, autonomous driving.**

I. Introduction

RECENT years have witnessed great advancement of autonomous driving technologies. To enable wide applications of autonomous vehicles, accurate and reliable moving-obstacle detection or segmentation in real dynamic traffic environments is an essential capability. This is because the perception of moving obstacles (e.g., moving cars or walking pedestrians) provides critical information for downstream tasks, such as motion planning [1], [2], decision making [3], [4], [5], and visual localization [6], [7], [8] in dynamic traffic environments.

Moving-obstacle segmentation refers to segmenting obstacles and discriminating their motion states (i.e., static or moving). Many existing methods segment moving obstacles in 2-D perspective-view images or 3-D LiDAR point clouds. Some of them [9], [10] are offline methods using a whole sequence of images, and some require prior-known information, such as the manually-labeled mask for the first image. Note that this work focuses only on online methods as well as methods that do not require manually-labeled masks during inference. Compared with segmentation on front-view images, bird-eye-view (BEV) is more straightforward since many downstream tasks (e.g., motion planning or prediction) work on such maps [11], [12]. To achieve moving-obstacle segmentation in BEV, we can use existing 3-D LiDAR point cloud-based methods [13], [14], [15], [16] to first get segmentation results in point clouds, and then project the results into BEV space. However, the major issue of this method is that the projected BEV maps are generally sparse due to the sparsity of LiDAR point clouds. The sparse BEV maps have many holes and gaps, making them hard to be directly used by downstream tasks. LiDAR sensors also suffer from high costs and computational burdens.

To provide a solution to the above issues, this paper proposes to generate dense BEV maps using images from low-cost visual cameras. Our network takes as input visual images captured by six surrounding cameras mounted on the ego-vehicle and directly generates the dense moving-obstacle segmentation maps in BEV. The reason for adopting six cameras in this work is to leverage the complementary fields of view (FOV), allowing our method to generate 360° ego-centric BEV maps. Note that in some literature, *moving obstacle* is also termed as *moving object*. Since this paper focus on traffic environments for autonomous driving research, the term *moving obstacle* would be more appropriate. In the following text, we do not discriminate them. Moreover, there exist some methods using visual images to generate semantic BEV maps [17], [18], [19], [20]. The major difference between our method and these methods is that we discriminate the motion states (e.g., static or moving) for obstacles, but these methods only segment obstacles at the semantics level (no motion state). In addition, some methods [21], [22], [23] can

Received 7 March 2023; revised 24 November 2023 and 6 April 2025; accepted 7 May 2025. Date of publication 3 June 2025; date of current version 1 July 2025. This work was supported in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515010116 and in part by the City University of Hong Kong under Grant 9610675. The Associate Editor for this article was S. Sacone. *(Corresponding author: Yuxiang Sun.)*

Shiyu Meng is with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: shiyu.meng@connect.polyu.hk).

Yuxiang Sun is with the Department of Mechanical Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong (e-mail: yx.sun@cityu.edu.hk).

Digital Object Identifier 10.1109/TITS.2025.3570058

only produce cone-shaped BEV maps due to the use of visual information from one camera view. The main contribution of this paper is a novel on-line method to produce dense moving-obstacle segmentation results in BEV. Our method only needs the images captured at the current and previous moments. Images captured at future moments are not required. To the best of our knowledge, this is the first work that produces moving-obstacle segmentation results in BEV using sequential images from multiple vehicle-mounted cameras. In addition, our method incorporates assisted semantic information into the network by proposing a movable-obstacle segmentation auxiliary task, and shows that the semantic information of movable obstacles could further benefit the moving-obstacles segmentation performance. Our code is open-sourced.[1] The contributions of this work are summarized as follows:

1) We propose a novel end-to-end framework that directly generates dense moving-obstacle segmentation results in BEV space.
2) We propose a geometry-aware BEV feature representation method that incorporates the semantic prior, camera intrinsic/extrinsic parameters, and attention module.
3) We introduce an auxiliary task, class-agnostic semantic segmentation for movable obstacles to further enhance our moving-obstacle segmentation performance.
4) We evaluate our network on both the large-scale nuScenes [24] and Lyft [25] datasets. The results demonstrate the superiority of our designed network.

The rest of this paper is organized as follows. Section II reviews the related work. Section III details our proposed network. Section IV presents the experimental results and discussions. The conclusions and future work are drawn in the last section.

II. Related Work

This section will summarize several related areas of our work, including semantic segmentation, moving obstacle segmentation, and BEV representations from images.

A. Semantic Segmentation

Semantic segmentation is a task that assigns each pixel in an image with a semantic class [26]. Many works have been done in this area. For example, Badrinarayanan et al. [27] proposed the encoder-decoder architecture in SegNet for semantic segmentation. Lo et al. [28] employed an asymmetric convolution structure and dilated convolution to solve the pixel classification task. Gao [29] used two parallel convolutional layers with different dilation rates to expand the FOV for boosting the performance of semantic segmentation. Xiao et al. [30] proposed BASeg, a CNN-based method consisting of semantic, boundary and aggregation streams, to learn boundary-aware features for semantic segmentation. Wang et al. [31] proposed a multi-view adapter-pair module to efficiently adapt pre-trained single-modality semantic segmentation networks to multi-modal networks.

B. Moving-Obstacle Segmentation

In contrast to semantic segmentation, the moving-obstacle segmentation task focuses on the obstacle state instead of detailed semantic classes. According to the sensors used, existing methods can be generally classified as vision-based methods and LiDAR-based methods. For vision-based methods, Sun et al. [32], [33] proposed to segment moving objects with traditional algorithms using an RGB-D camera. Vertens et al. [34] proposed a deep learning-based method to segment moving objects from front-view images and their corresponding optical flow information. Jain et al. [35] proposed an automatic segmentation of foreground prominent objects in videos combining the motion information. Liu and Wang [36] proposed a CNN-based motion segmentation method based on consecutive depth maps generated from LiDAR data. Siam et al. [37] proposed a two-stream CNN-based network to conduct object detection and motion segmentation task. For LiDAR-based methods, Sun et al. [13] proposed a sparse tensor-based end-to-end moving-obstacle segmentation network using point clouds from a 3-D LiDAR. Chen et al. [38] proposed an off-line moving object segmentation method with cluster and Kalman filter. Sun et al. [39] applied a CNN-based network with the meta-kernel convolution to range images from point clouds and applied a back projection to get point-wise predictions. Kim et al. [40] proposed a solution to segment moving obstacles given a sequence of range images from point clouds and project the outputs back to LiDAR space.

C. BEV Perception From Visual Images

Traditional BEV perception methods mainly rely on the Inverse Perspective Mapping (IPM) algorithm [41]. The major issue of the IPM algorithm is the flat-road assumption, making the methods not suitable for uneven road environments. Deep learning-based end-to-end methods can avoid this issue. Lu et al. [21] proposed an end-to-end approach to learn the monocular occupancy BEV maps with a monocular camera. Roddick and Cipolla [42] proposed an end-to-end deep learning solution via a Bayesian occupancy grid framework. Can et al. [18] proposed a transformer-based deep neural network to transform the perspective view to BEV. Dwivedi et al. [43] proposed a deep neural network to produce semantic BEV maps with lifted 2-D semantic features. Zhou and Krähenbühl [44] designed an attention-based model and implicitly learned a mapping from individual camera views into map-view representation.

Due to the practical use of BEV maps, scene understanding in BEV using images has recently gathered significant attention. However, most current research focuses mainly on semantic segmentation [45], [46], [47]. The category classification, not the status classification, is the problem to be solved. Meanwhile, these approaches take previous and future information as input, which is not suitable for real autonomous driving applications.

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

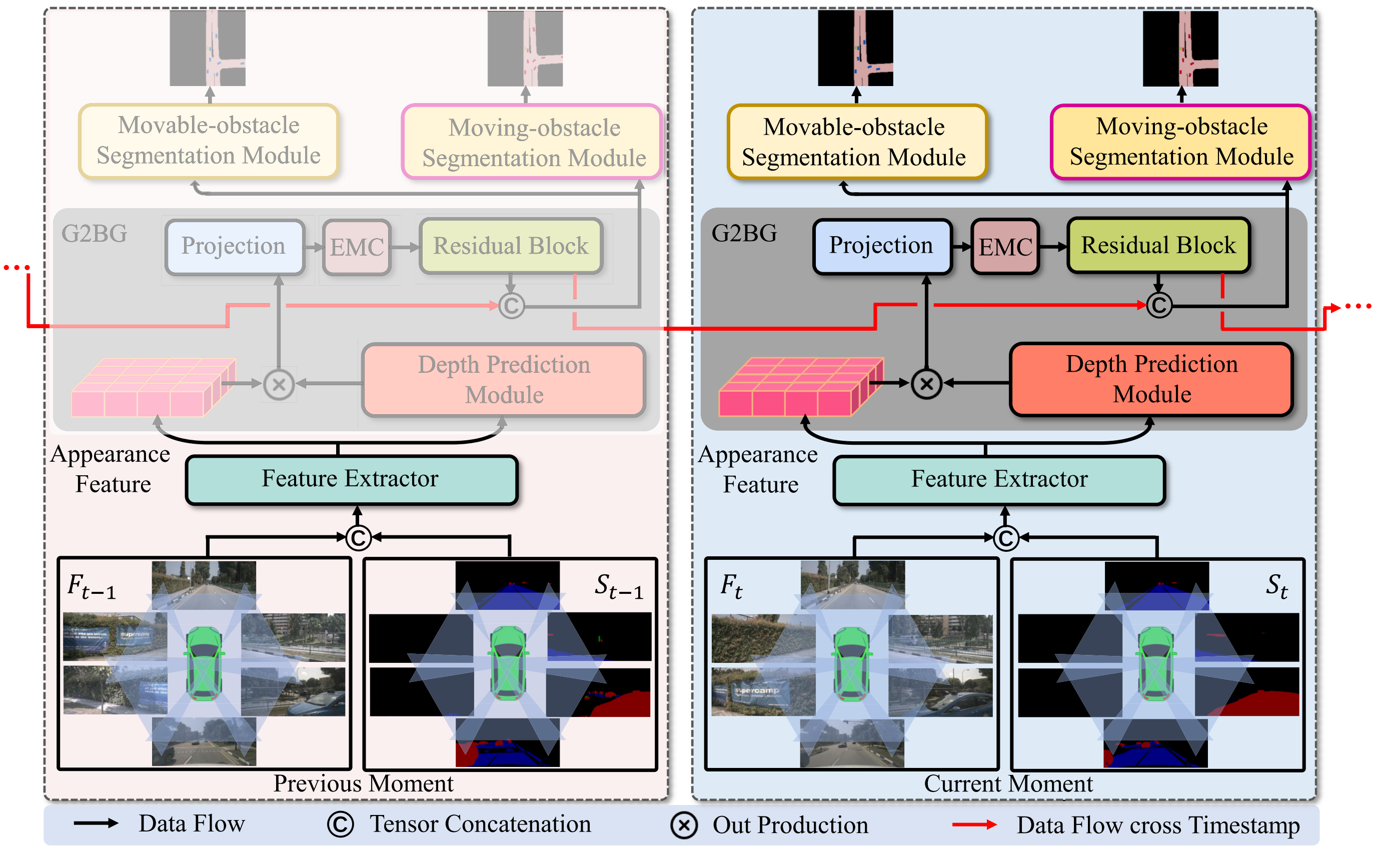

Fig. 1. The overall structure of our proposed network. At each timestamp, the collected six visual images are displayed on the left, and their corresponding semantic masks are displayed on the right. The auxiliary task and the main task respectively mean the movable-obstacle segmentation task and the moving-obstacle segmentation task. F and S respectively represent the visual images and corresponding semantic prior information. The appearance feature is the feature extracted by the feature extractor. EMC represents ego-motion compensation. The 🟥 color in the moving-obstacle segmentation map represents moving obstacles (i.e., moving vehicles and pedestrians). The 🟦 color in the movable-obstacle segmentation map represents the movable obstacles. The ■ and 🟩 colors respectively represent drivable areas and the ego-vehicle. Note that our network does not segment them. They are just displayed for visualization.

III. The Proposed Network

A. The Overall Architecture

Fig. 1 shows the structure of our network Semantic-MoSeg. It can be seen that it mainly consists of four modules: feature extractor, geometry-guided BEV generation (G2BG) module, moving-obstacle segmentation module, and movable-obstacle segmentation module (i.e., the auxiliary task). Our network takes as input two sets of six images from the surrounding cameras mounted on the ego-vehicle that are captured at different moments (the time interval is fixed) and directly generates moving-obstacle segmentation maps and movable-obstacle segmentation maps of the current moment.

As shown in Fig. 1, semantic segmentation is first performed on six input images from the surrounding cameras using [48] to generate semantic segmentation maps $S_t = \left\{S_t^1, \ldots, S_t^n\right\}$ at the current moment t, where $n \in \{1, \ldots, 6\}$. The segmentation maps are visualized in 3-channel color images. The semantic segmentation maps are adopted as our prior information. The input images and the corresponding semantic segmentation maps are concatenated to form 6-channel images. Secondly, a feature extractor is adopted to extract visual features from the 6-channel images from the current and previous moments. All the images share the same feature extractor module. Thirdly, the G2BG module transforms the perspective-view features to BEV and produces BEV feature maps for the consecutive two moments. Finally, the moving-obstacle segmentation module and the movable-obstacle segmentation module are employed to produce the respective segmentation results. The movable-obstacle and moving-obstacle segmentation maps are both with the size of $N_c \times H \times W$, where N_c is the number of classes, H and W represent height and width.

B. Feature Extractor

As aforementioned, the input data to the feature extractor are the concatenated 6-channel images from F_t and S_t at current and previous moments. We choose EfficientNet-B4 [49] as our backbone for the feature extractor due to its lightweight architecture. Specifically, we adopt EfficientNet-B4 with the random initialization scheme. We modify the input channel number of the first layer so that the feature extractor can take as input the concatenated images. The concatenated images are finally downsampled with a factor of 8.

C. Geometry-Guided BEV Generation (G2BG)

The G2BG module is designed to generate BEV features. We follow [17] to transform perspective-view features to BEV feature maps by predicted depth distributions. Firstly, depth

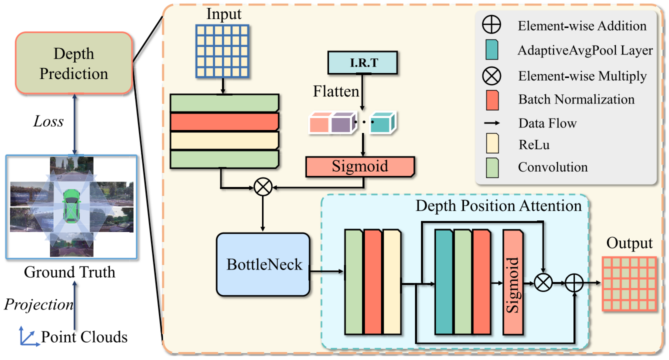

Fig. 2. The structure of our proposed depth prediction module. The input of this module is the output of the feature extractor module. I.R.T represents the camera intrinsic matrix (3 × 3), extrinsic rotation matrix (3 × 3), and extrinsic translation vector (3 × 1). The ground truth for depth prediction is the sparse depth from LiDAR point clouds, which are overlaid on the RGB images for visualization. Please zoom in to see details.

prediction is conducted based on the features produced by the feature extractor, which is shown in Fig. 2. We simply flatten the camera intrinsic matrix (3 × 3) and extrinsic matrix (3 × 3 rotation and 3 × 1 translation) into a vector with the length of 21. To further enhance the depth prediction, we design a depth position attention (DPA) module. The size of the convolution kernel in the DPA module is 3 × 3. Specifically, the sparse depth data from the LiDAR point clouds is utilized as the supervision signal to train the depth prediction network. With the predicted depth, we project the perspective-view features to BEV features.

Secondly, we conduct ego-motion compensation (EMC) on the BEV features. Since the differences between the current images and the previous images are caused by both the ego-vehicle motion and object motion. We adopt the vehicle pose between the two moments to conduct EMC. The idea is to transform all the previous features into the coordinate system at the current moment.

Finally, the coarse BEV feature maps are further refined by exploiting the temporal information between the two moments. A residual module is proposed to find the difference between the coarse BEV feature maps at the two moments:

$$\hat{bev_t} = bev_t + bev_t - bev_{t-1}, \tag{1}$$

where bev_t is the output of the residual block, $\hat{bev_t}$ is the difference. The bev_t is doubled to avoid $\hat{bev_t}$ to be zero. Then, we concatenate the residuals calculated from the previous moment as the output of the G2BG module. The idea behind the residual module is that the static parts would become smaller after subtraction, which in turn amplifies the differences between the moving part and static part.

D. Moving-Obstacle Segmentation

We design the moving-obstacle segmentation module based on a modified DenseNet [50], atrous spatial pyramid pooling, and skip connections. At the end of the module, a moving-obstacle segmentation head is added, which is a 2-D convolutional layer, to provide dense pixel-wise prediction. The BEV map range is $100m \times 100m$ with a unit length of $0.5m$. So, the resolution of the BEV map is 200 × 200.

E. Movable-Obstacle Segmentation

The movable-obstacle segmentation module is designed as the auxiliary task, to endow the network with the ability to inherently and implicitly learn what kind of obstacles are possible to move. The class-agnostic movable-obstacle segmentation task here is a binary segmentation task (i.e, movable or non-movable). For example, both pedestrians and parked cars are considered as the same class (i.e., movable objects). The task enhances our moving-obstacle segmentation performance since moving objects should be movable objects (e.g., pedestrians, vehicles). Note that movable is a concept of semantics. It does not involve motion states. We adopt the same network as that of the moving-obstacle segmentation module and share the same parameters between them. The only difference is that we replace a movable-obstacle segmentation head with the moving-obstacle segmentation head to achieve the movable-obstacle segmentation.

F. Loss Functions

We use the binary cross entropy (BCE) loss for both moving and movable segmentation, as well as the depth distribution prediction. The three losses are denoted as $\mathcal{L}_{moving}$, $\mathcal{L}_{movable}$, and $\mathcal{L}_{depth}$. We follow [51] to compute the depth loss. The total loss is:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{moving} + \beta \cdot \mathcal{L}_{movable} + \gamma \cdot \mathcal{L}_{depth}, \tag{2}$$

where α, β, and γ are learnable parameters to weight the three losses.

IV. Experimental Results and Discussions

A. Datasets

To evaluate the performance of the proposed method, we conduct experiments on two public datasets: nuScenes [24] and Lyft [25]. The nuScenes dataset includes 1,000 scenes with visual images from six vehicle-mounted cameras and point clouds from a 3-D LiDAR sensor. Among the scenes, 850 scenes are given ground-truth annotations. The moving-obstacle ground-truth labels are generated by filtering the attributes and projecting the provided ground-truth bounding boxes into BEV to obtain 2-D polygons. The moving obstacles in nuScenes have the attributes of `pedestrian.moving` and `vehicle.moving`. All the images in the same scene are used either for training, validation, or testing. We randomly split the 850 scenes into training (550 scenes), validation (150 scenes), and testing (150 scenes). The Lyft dataset also provides multi-view information captured by visual sensors mounted on the roof of the ego-vehicle. There are 180 scenes with ground-truth annotations. We randomly split 36 scenes for testing.

B. Implementation Details

We implement our network with PyTorch and the PyTorch-Lightning [52] library. The network is trained on NVIDIA RTX 3090. The resolution of the input images is resized to 224 × 480 for the experiments. The proposed network

TABLE I

THE RESULTS OF ABLATION STUDY ON DIFFERENT FEATURE EXTRACTORS. *Movable* REPRESENTS MOVABLE-OBSTACLE SEGMENTATION. *Moving* REPRESENTS MOVING-OBSTACLE SEGMENTATION

Extractor	Movable		Moving	
	IoU %	Precision %	IoU %	Precision %
Efficient-B0	37.62	49.97	32.54	43.74
Efficient-B1	38.60	52.08	33.33	45.12
Efficient-B2	38.45	51.25	33.72	46.91
Efficient-B3	39.49	53.26	34.76	49.06
Efficient-B4	39.57	53.08	35.08	48.61

processes a sequence of 6 camera images and outputs BEV maps with 200×200 resolution at $50cm$ unit length in both the x and y directions. If not stated differently in the experiments, the time interval for our experiment is the time between two adjacent key frames.

The Adam optimizer with decoupled weight decay [53] are adopted for training. The initial learning rate is set as 10^{-3}. We adopt the MultiStepLR to decay the learning rate during training. In particular, we train all the experiments under mixed precision. The training data are randomly shuffled before each epoch. Since we perform the shuffle operation on the whole set of consecutive frames, the order of the image sequences is not influenced.

C. Evaluation Metrics

We adopt two evaluation metrics for quantitative evaluation: Precision and Intersection-over-Union (IoU) [54]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{3}$$

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively. Note that our metrics are calculated with respect to the ground truth at the current moment.

D. Ablation Study

We create several variants of our network to verify the effectiveness of our design. We train all the variants up to 30 epochs and report the best results on the nuScenes dataset for moving-obstacle segmentation.

1) Ablation on Feature Extractor: The EfficientNet has variants from B0-B7, where B5-B7 contains more parameters, which improves network performance but also increases computational cost. To trade off performance and computational cost, we only conduct the ablations using EfficientNet B0-B4. From Tab. I, we can see that the variant with EfficientNet-B4 provides the best performance. So, unless otherwise specified, EfficientNet-B4 is adopted as the feature extractor for our experiments.

2) Ablation on Ego-Motion Compensation: This ablation study is to demonstrate that introducing ego-motion compensation could increase the moving-obstacle segmentation performance. Tab. II shows the results. We can see that with the ego-motion compensation, the IoU of the moving-obstacle segmentation has been improved with 0.85%, and the precision has been improved with 0.89%.

3) Ablation on Geometry Awareness: This ablation study is to demonstrate the benefits of introducing the camera's intrinsic/extrinsic parameters. The results are shown in Tab. II. It can be illustrated that the camera parameters are beneficial to the moving-obstacle segmentation performance. From Variant-6, we can see that with the intrinsic and extrinsic parameters, the IoU of our method has been improved by 0.34%. It also improves precision by 1.29% in the movable-obstacle segmentation.

4) Ablation on Residual Block: This ablation study aims to demonstrate the effectiveness of the residual block. Tab. II shows the results. It can be seen that the residual layer is positive on moving-obstacle segmentation based on the residuals of the time-series feature maps. From Variant-5, we can find that with the residual operation, the IoU of our network has been improved by 0.99%, and the precision has been improved by 1.24%. It can be observed that with the amplified difference between the static and moving obstacles, it is easier for the network to find moving obstacles.

5) Ablation on Depth Supervision: This ablation study aims to demonstrate the depth supervision performance. The results are shown in Tab. III. As aforementioned, the spare depth ground truth is obtained from the LiDAR point cloud. It can be seen that with the depth supervision, the IoU of our network has been improved by 1.14%, and the precision of segmenting moving obstacles has been improved by 1.13%. This indicates that depth supervision could further benefit the moving-obstacle segmentation performance.

6) Ablation on Auxiliary Task: This ablation study is designed to demonstrate whether adding the auxiliary task, that is, the movable-obstacle segmentation, is helpful for boosting the performance of our moving-obstacle segmentation. The experimental results are displayed in Tab. III. We can see that the auxiliary task to learn the features from movable obstacles in the surroundings is helpful in improving moving-obstacle segmentation performance. This is reasonable because moving obstacles have to be movable.

7) Ablation on Detection Ranges: This ablation study is designed to demonstrate the robustness and performance of our network on different detection ranges. Here, we have three different settings: $100m \times 100m$ resolution at $50cm$ unit length, $80m \times 80m$ resolution at $40cm$ unit length, and $100m \times 50m$ resolution at $25cm$ unit length. Tab. IV shows the results. It can be seen that our network performs robustly at different detection ranges.

8) Ablation on Loss Weights: This ablation study is to verify whether using learnable weights for the losses is helpful. The results are displayed in Tab. V. We can see that the IoU results with the learnable weights are better than those without the learnable weights. Setting the weights equal to each other is still effective but sub-optimal. Generally, it is helpful to use the learnable weights.

9) Ablation on Semantic Prior Information: This ablation study is to demonstrate the benefit brought by the semantic

TABLE II

THE RESULTS OF ABLATION STUDY ON DIFFERENT MODULE COMPONENTS IN G2BG. *Geometry-Aware* MEANS TO USE THE CAMERA PARAMETERS TO GUIDE THE GENERATION OF THE BEV FEATURES. THE RESULTS DEMONSTRATE THE EFFECTIVENESS OF DIFFERENT COMPONENTS OF THE G2BG MODULE FOR THE MOVING-OBSTACLE AND MOVABLE-OBSTACLE SEGMENTATION TASKS

Variants No.	Timestamp	Geometry-aware	Pose	Residual Block	Movable		Moving	
					IoU %	Precision %	IoU %	Precision %
Variant-1	2	✗	✗	✗	38.03	51.37	31.09	43.26
Variant-2	2	✓	✗	✗	38.75	51.83	32.36	44.30
Variant-3	2	✗	✓	✗	39.41	53.37	33.38	45.80
Variant-4	2	✗	✗	✓	37.55	49.31	33.03	43.79
Variant-5	2	✓	✓	✗	39.78	53.99	34.09	47.37
Variant-6	2	✓	✗	✓	39.33	52.95	34.23	47.72
Variant-7	2	✗	✓	✓	39.80	53.93	34.74	47.32
Variant-8	2	✓	✓	✓	39.57	53.08	35.08	48.61

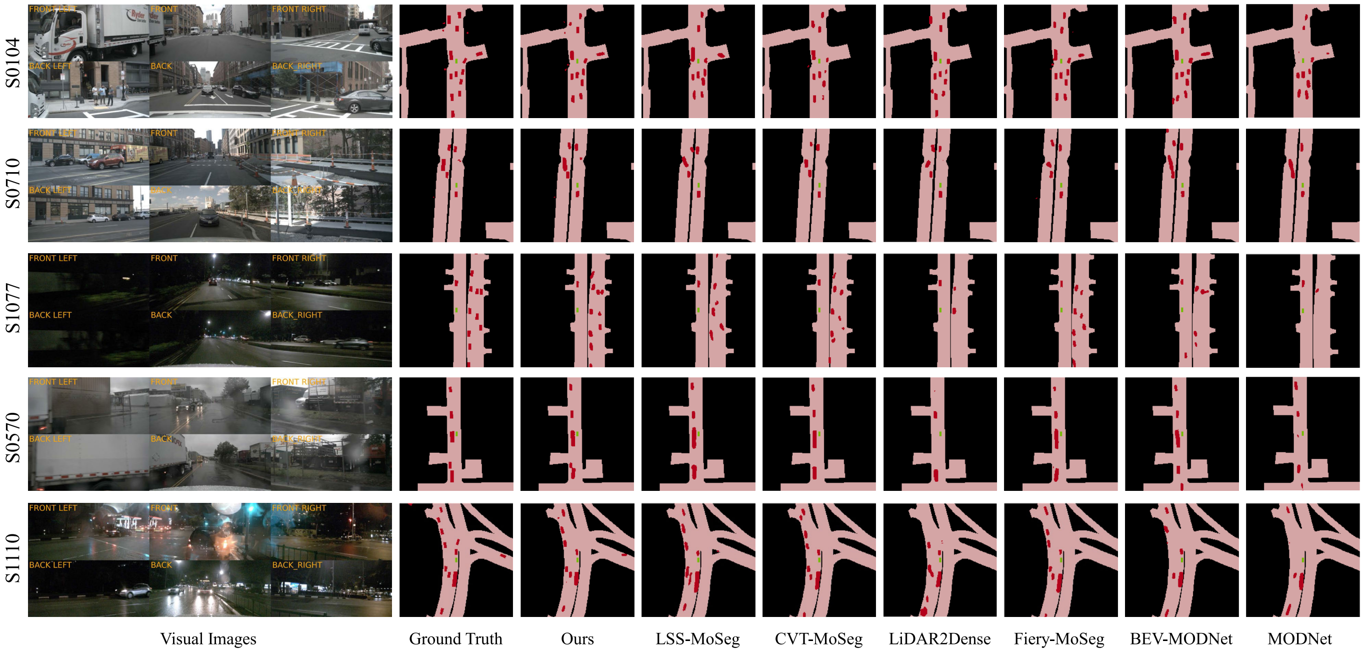

Fig. 3. Sample qualitative demonstrations for moving-obstacle segmentation under different weather and lighting conditions: sunny, rainy, cloudy, and nighttime. The colors 🟥, 🟩, and ▨ respectively represent moving obstacles, ego-vehicle, and drivable areas. The six visual images are from the front camera, front-left camera, front-right camera, back camera, back-left camera, and back-right camera. Note that the ego-vehicle and drivable area are only used for visualization. Our network does not segment them. The left texts show the ID of the scenes, where *S* is short for *scene*.

TABLE III

THE RESULTS OF ABLATION STUDY ON THE DEPTH SUPERVISION AND AUXILIARY TASK. *Movable* AND *Moving* REPRESENT THE MOVABLE-OBSTACLE AND MOVING-OBSTACLE SEGMENTATION TASKS, RESPECTIVELY

Supervision	Sub-task	Movable		Moving	
		IoU %	Precision %	IoU %	Precision %
✗	✗	–	–	32.16	44.10
✗	✓	38.84	51.38	33.94	47.48
✓	✓	39.57	53.08	35.08	48.61
✓	✗	–	–	33.78	46.48

TABLE IV

THE RESULTS OF ABLATION STUDY ON DIFFERENT RANGES. *Movable* AND *Moving* REPRESENT THE MOVABLE-OBSTACLE AND MOVING-OBSTACLE SEGMENTATION TASKS, RESPECTIVELY

Timestamp	Range	Movable		Moving	
		IoU %	Precision %	IoU %	Precision %
2	(-25m, 25m, 0.25m)	53.05	68.51	45.05	59.70
2	(-40m, 40m, 0.4m)	40.68	52.84	35.91	47.04
2	(-50m, 50m, 0.5m)	39.57	53.08	35.08	48.61

prior information. The results are displayed in Tab. VI. The variant Visual-only refers to using the captured visual multi-view images as input. The variant Prior-only refers to using the semantic-prior information as input. The variant Element-wise summation refers to using the addition between visual multi-view images as input. We can see that the moving-obstacle segmentation performance is gradually enhanced as the input information is gradually enriched. Moreover,

TABLE V

THE RESULTS OF ABLATION STUDY ON LOSS WEIGHTS. *Movable* AND *Moving* REPRESENT THE MOVABLE-OBSTACLE AND MOVING-OBSTACLE SEGMENTATION TASKS, RESPECTIVELY. ✓ AND ✗ RESPECTIVELY REPRESENT WITH AND WITHOUT THE LEARNABLE WEIGHTS. FOR ✗, WE SIMPLY SET THE WEIGHTS EQUAL TO EACH OTHER

Timestamp	Weight	Movable		Moving	
		IoU %	Precision %	IoU %	Precision %
2	✗	39.45	53.73	34.71	48.17
2	✓	39.57	53.08	35.08	48.61

TABLE VI

THE RESULTS OF ABLATION STUDY ON SEMANTIC PRIOR INFORMATION. *Movable* AND *Moving* REPRESENT THE MOVABLE-OBSTACLE AND MOVING-OBSTACLE SEGMENTATION TASKS, RESPECTIVELY

Timestamp	Input Type	Movable		Moving	
		IoU %	Precision %	IoU %	Precision %
2	Visual-only	39.14	53.52	32.44	44.60
2	Prior-only	37.98	51.13	31.98	44.31
2	Element-wise addition	39.13	53.16	34.30	47.60
2	Semantic-MoSeg (ours)	39.57	53.08	35.08	48.61

TABLE VII

COMPARATIVE RESULTS OF DIFFERENT METHODS ON THE NUSCENES DATASET. *Moving* REPRESENTS MOVING-OBSTACLE SEGMENTATION. THE TABLE DEMONSTRATES THE SUPERIORITY AND EFFECTIVENESS OF OUR METHOD

Timestamp	Baseline Type	Moving	
		IoU (%)	Precision (%)
2	BEV-MODNet	22.17	30.07
2	SMSnet	12.18	14.97
2	MODNet	13.68	17.14
2	LSS-MoSeg	24.60	37.34
2	CVT-MoSeg	27.41	37.81
2	Fiery-MoSeg	29.89	40.08
2	LiDAR2Dense	25.13	47.42
2	Semantic-MoSeg (ours)	35.08	48.61

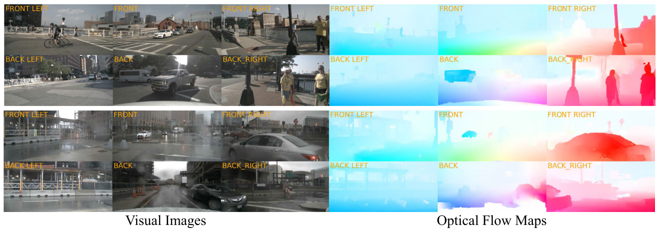

Fig. 4. The generated optical flow from the visual images of the six cameras. The left part shows the multi-view visual images, and the right part shows the corresponding optical flow maps.

feature concatenation achieves better performance compared to element-wise summation. The reason may be that the concatenation is helpful for our method to adaptively learn feature maps. The incorporation of semantic information could make the network segment moving obstacles more easily.

In summary, our network generally outperforms the other variants, which demonstrates the effectiveness of our design. With different strategies, the different components boost the performance in terms of IoU and precision. This demonstrates that it is beneficial to incorporate all the proposed components.

E. Comparative Experiments

In this section, we compare our method with the existing methods (i.e., BEV-MODNet [55], SMSnet [34], and MODNet [37]), and create several baselines (i.e., LSS-MoSeg, CVT-MoSeg, and Fiery-MoSeg) for comparison. For all the compared methods, the output layers are modified to generate moving-obstacle segmentation maps in BEV. The detailed descriptions for the compared methods are listed as follows:

- **BEV-MODNet**: This method is proposed for BEV moving object detection based on the front-view images. We first generate the optical flow for the multi-view visual images, as shown in Fig. 4. Then we re-implement the model based on the architecture and adopt the multi-view visual images as inputs.
- **SMSnet**: This method is designed to detect moving objects in a perspective view based on two consecutive time-stamp inputs. We adopt the multi-view visual images as inputs and project the detection results via the IPM algorithm. We train this baseline from scratch.
- **MODNet**: The MODNet method is proposed for perspective-view detection based on the visual image and its corresponding optical flow. We generate the optical flow for the visual images and project the segmentation results to BEV via the IPM algorithm. We train this baseline from scratch.
- **LSS-MoSeg**: This baseline is based on the LSS method [17]. The input of LSS is the monocular images at a single moment. Since moving-obstacle segmentation requires sequential data, we change the time interval length to 2 moments. We name this baseline as LSS-MoSeg.
- **CVT-MoSeg**: This baseline is based on the CVT approach [17]. We change the time interval of the CVT method to 2. We name this baseline as CVT-MoSeg.
- **Fiery-MoSeg**: This baseline is based on the Fiery method [45]. The original Fiery is for semantic segmentation with sequential data. Since the future information could not be used for online applications, such as autonomous driving. So, we omit the sub-module using future features in [45]. We set the time interval length to 2. We name this baseline as Fiery-MoSeg.
- **LiDAR2Dense**: This baseline is based on PointPillars [56]. It adopts the same moving-obstacle segmentation head as ours to generate dense predictions. Here, we only use the 3-D coordinates of the point clouds as input. The other information, such as intensity, is discarded. Since our output is dense BEV maps for moving obstacles, we name this baseline as LiDAR2Dense.

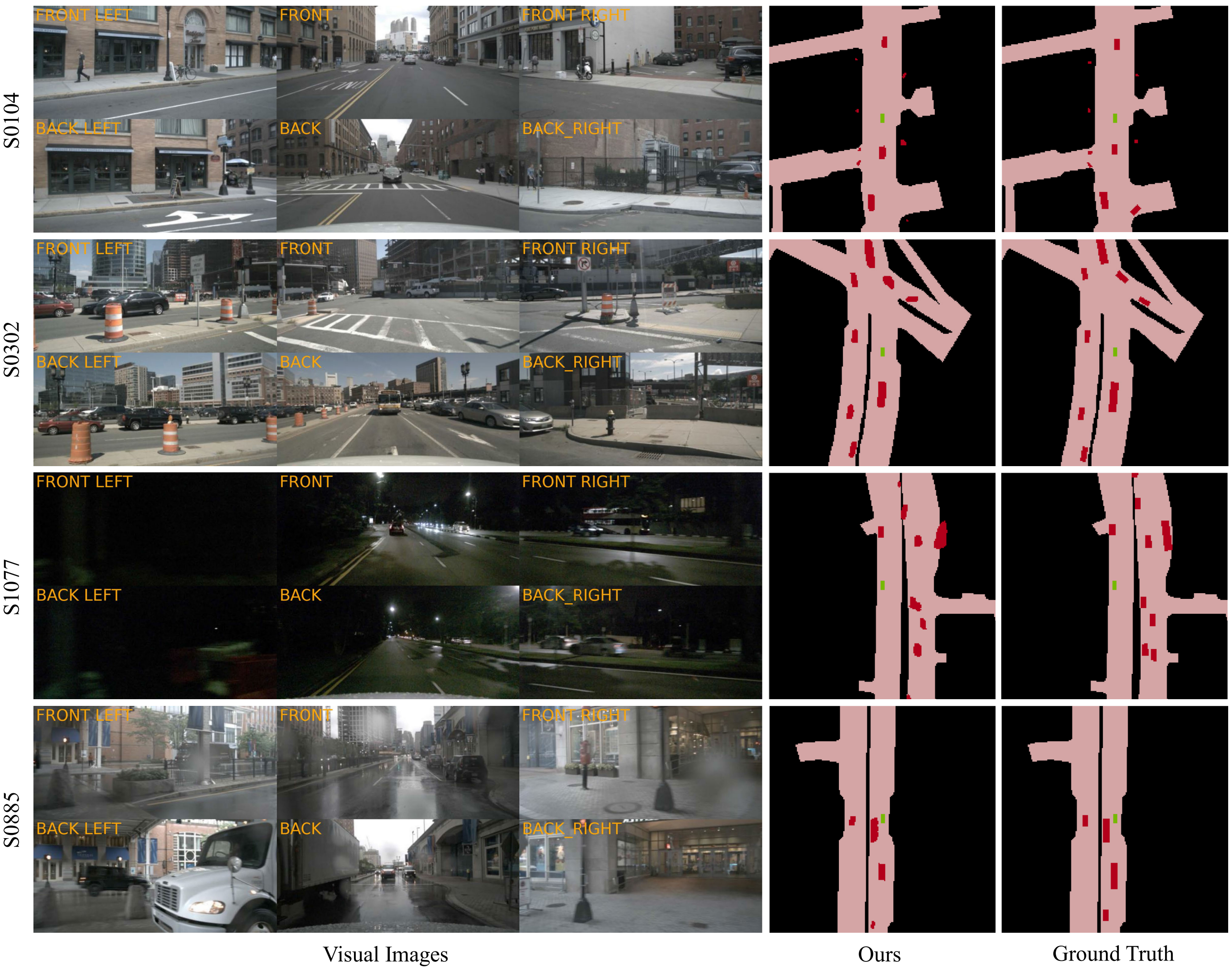

Fig. 5. Sample qualitative demonstrations of our network for moving-obstacle segmentation on the nuScenes dataset. The first, second, third, and fourth rows respectively show the cloudy, sunny, night, and rainy conditions. The colors 🟥, 🟩, and ▨ respectively represent moving obstacles, ego-vehicle, and drivable areas. The six visual images are from the front camera, front-left camera, front-right camera, back camera, back-left camera, and back-right camera. Note that the ego-vehicle and drivable area are only used for visualization. Our network does not segment them. The left texts show the ID of the scenes, where *S* is short for *scene*.

As shown in Tab. VII, we can see that our Semantic-MoSeg achieves the best performance compared with all the other methods, which demonstrates our superiority. Fig. 3 qualitatively demonstrates sample comparative results for moving-obstacle segmentation. We can see that the segmentation performance of our Semantic-MoSeg generally outperforms the other methods. This could be attributed to the effective representation of multi-view visual features and the integration of the auxiliary movable-obstacle features. The LSS-MoSeg, CVT-MoSeg, and Fiery-MoSeg baselines lack movable-obstacle features, which are helpful for moving-obstacle segmentation. Furthermore, the SMSnet and MODNet methods require post-projection and optical flow calculation, introducing intermediate errors that might affect segmentation performance. Specifically, the first row shows a cloudy scenario. We can see that there are moving trucks, cars, and pedestrians in the scene. Our network is the only one that can segment both moving vehicles and pedestrians. The second row shows a sunny scenario. Our network is the only one that segments moving cars and pedestrians under BEV space. For the remaining comparative methods, they both miss the pedestrians crossing the road. The third row shows a nighttime scenario. Under such light conditions, streetlights and vehicle headlights share similar visual appearances. We can see that our results are more complete than the other baselines. The other baselines all miss some segmentation to some degree, especially when objects are far from the ego vehicle. The fourth row shows a rainy scenario. Under such weather conditions, although the lens of the cameras are blurred, our network still detects all the moving vehicles. The last row shows a scenario with on-coming and nearby headlights under a rainy night condition. Although the light is weak and there is interference from water ripples, our network can still accurately segment the moving obstacles.

F. Qualitative Demonstrations

We also demonstrate some qualitative results of our method on both moving-obstacles and movable-obstacles segmentation, which are displayed in Fig. 5 and Fig. 6. We can see that our Semantic-MoSeg generalizes well to unseen

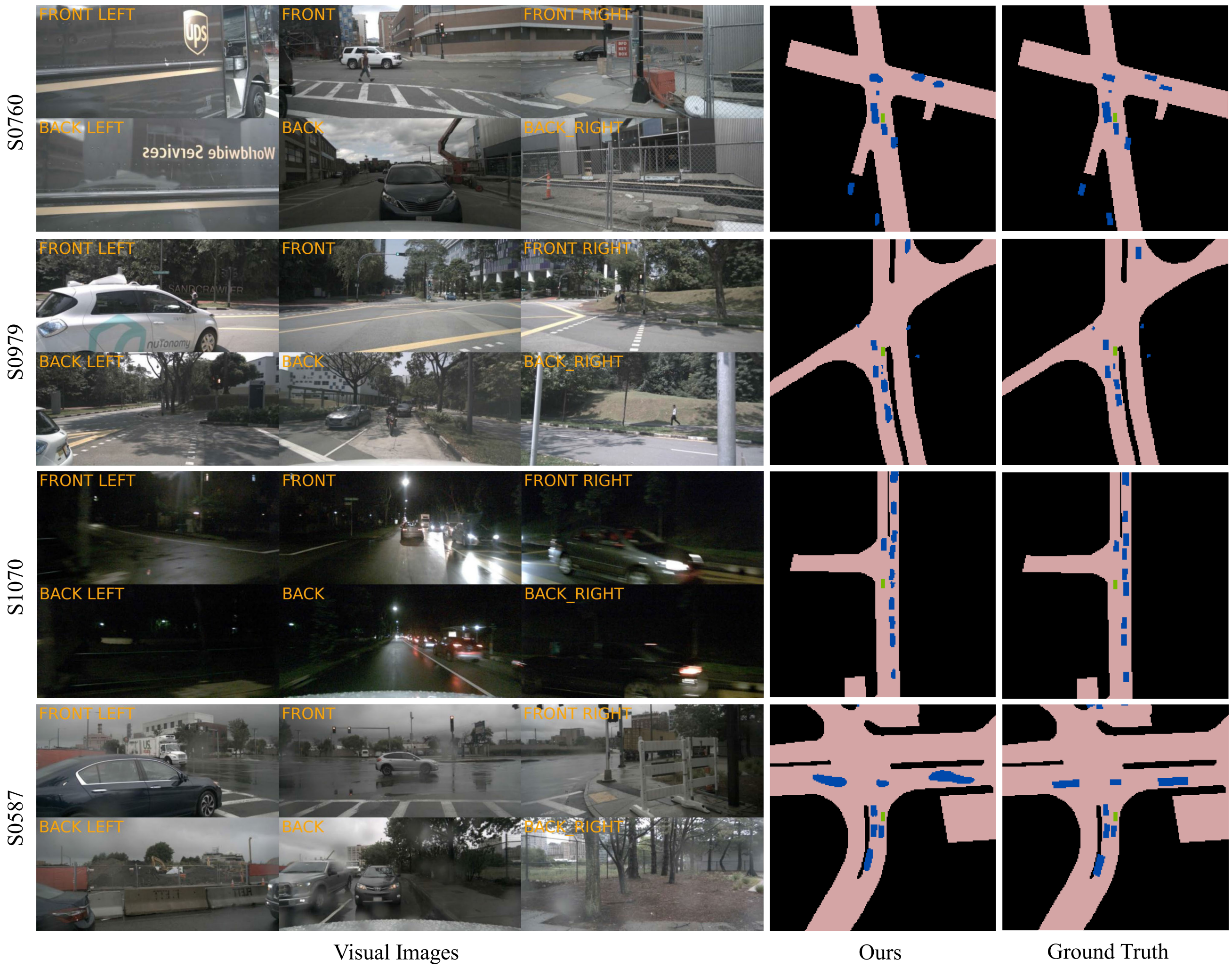

Fig. 6. Sample qualitative demonstrations of our network for auxiliary task, movable-obstacle segmentation, on the nuScenes dataset. The first, second, third, and fourth rows respectively show the cloudy, sunny, night, and rainy conditions. The colors 🟦, 🟩, and 🟥 respectively represent movable obstacles, ego-vehicle, and drivable areas. The six visual images are captured from the front camera, front-left camera, front-right camera, back camera, back-left camera, and back-right camera. Note that the ego-vehicle and drivable area are only used for visualization. Our network does not segment them. The left texts show the ID of the scenes, where *S* is short for *scene*.

TABLE VIII

ROBUSTNESS EVALUATION OF OUR NETWORK ON THE LYFT DATASET. *Movable* AND *Moving* REPRESENT THE MOVABLE-OBSTACLE AND MOVING-OBSTACLE SEGMENTATION TASKS, RESPECTIVELY

Timestamp	Input Type	Movable		Moving	
		IoU %	Precision %	IoU %	Precision %
2	Semantic-MoSeg (ours)	45.11	60.28	43.19	55.43

environments under various weather and lighting conditions, including sunny, rainy, cloudy, and nighttime.

G. Robustness Evaluation

1) Robustness on Different Datasets: To demonstrate the robustness of our method, we evaluate our network on the other public dataset, Lyft. We get the moving-obstacle ground truth by filtering and merging the annotated attributes in the same way as the nuScenes dataset. The range for moving-obstacle segmentation is also $100m \times 100m$. Tab. VIII displays the experimental results. We can see that compared to the results on the nuScenes dataset, our network exhibits robust performance on the Lyft dataset. Meanwhile, our method can also exhibit robustness on our auxiliary movable-obstacle segmentation task.

2) Robustness on Different Views: To validate the robustness of our method on different views, we randomly remove an image from an viewpoint among the six viewpoints during test, and then apply the saved model weights to the remaining images. Fig. 7 shows the results. We can see that the performance of our network varies when removing different views, but generally is robust when front-left, front-right, back-right, or back-right is removed. Our method performs best in terms of mean IoU (mIoU) from the front-left view, and performs best in terms of mean Precision (mPre) from the back-right view. When front or back view is removed, the performance of our network is degraded. We conjecture the reason is that the front view encodes more

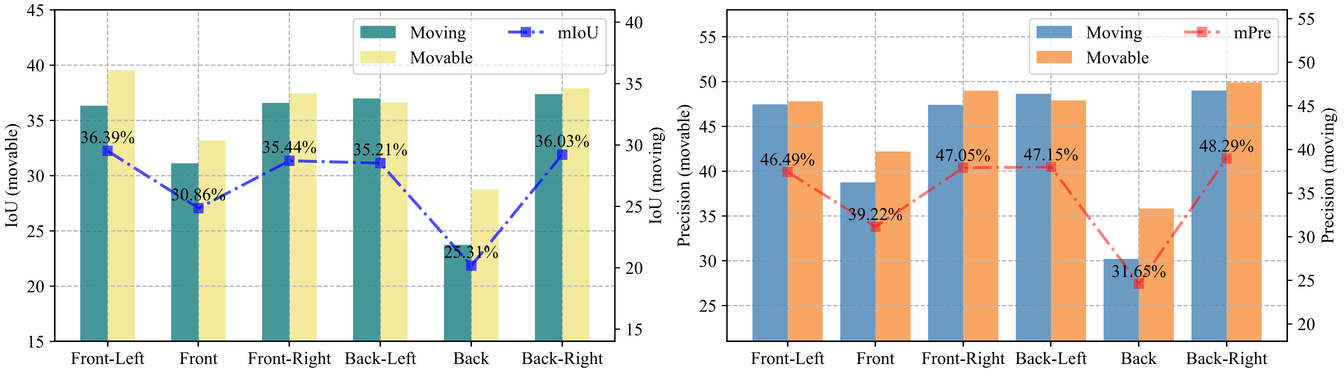

Fig. 7. Robustness evaluation of our network with different camera views on the nuScenes dataset. The results demonstrate our stability and robustness. *Movable* and *Moving* represent the movable-obstacle and moving-obstacle segmentation tasks, respectively. mPre and mIoU are the mean values of precision and IoU, respectively.

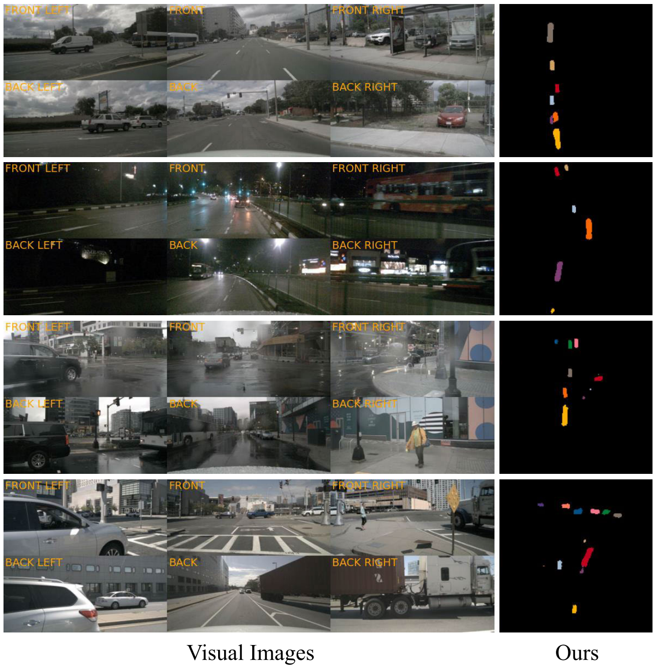

Fig. 8. Sample qualitative demonstrations for moving-obstacle instance segmentation on the nuScenes dataset. The rows from top to bottom respectively show cloudy, night, rainy, and sunny conditions.

information and most road environment information appears in the front region. The back view is just the opposite of the front view, so it also encodes more information. In general, we can see that the mIoU and mPre results have a similar trend of performance variations.

H. Application Study

In this section, we demonstrate one application of our proposed network, that is, moving-obstacle instance segmentation. This application makes full use of the moving-obstacles segmentation results. We set the time interval as 2 adjacent keyframes. A simple moving-obstacle instance segmentation head with several convolutional layers is added to our model to distinguish different moving obstacles at the instance level. We retrain the model, and the qualitative demonstrations are shown in Fig. 8. We can see that our network is able to produce moving-obstacle BEV maps at the instance level.

V. Conclusion and Future Work

In this paper, we proposed a novel end-to-end network for online moving-obstacle segmentation in BEV with multiple images captured at different moments. We solve the problem by projecting the perspective-view feature maps to BEV and incorporating camera intrinsic/extrinsic parameters and prior semantic information explicitly. Moreover, an auxiliary task, movable-obstacle segmentation, is introduced to further improve our moving-obstacle segmentation performance. Extensive experiments on the nuScenes and Lyft datasets demonstrate the robustness and superiority of our proposed approach in moving-obstacle segmentation. Despite the superiority of our proposed network, there still exist some limitations. Our proposed method currently relies only on visual cameras, which inherently limits the robustness to challenging scenarios, such as low illumination and adverse weather conditions. In the future work, we would like to incorporate multi-modal sensor data, such as LiDAR point clouds or thermal images, which may complement visual cues to enhance the segmentation robustness. In addition, we would like to explore the potential of moving-obstacle segmentation for other downstream tasks, such as panoptic segmentation and localization.

References

[1] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020.

[2] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, 2020, pp. 414–430.

[3] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 2, pp. 294–305, Jun. 2020.

[4] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2076–2087, Apr. 2021.

[5] P. Ke, Z. Yanxin, and Y. Chenkun, "A decision-making method for Self-driving based on deep reinforcement learning," in *Proc. J. Phys., Conf.*, 2020, vol. 1576, no. 1, Art. no. 012025.

[6] N. Radwan, A. Valada, and W. Burgard, "VLocNet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4407–4414, Oct. 2018.

[7] D. Zhang, J. Han, G. Cheng, and M. H. Yang, "Weakly supervised object localization and detection: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5866–5885, Sep. 2021.

[8] H. Yin, S. Li, Y. Tao, J. Guo, and B. Huang, "Dynam-SLAM: An accurate, robust stereo visual-inertial SLAM method in dynamic environments," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 289–308, Feb. 2023.

[9] P. W. Patil, K. M. Biradar, A. Dudhane, and S. Murala, "An end-to-end edge aggregation network for moving object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8149–8158.

[10] J. H. Giraldo, S. Javed, M. Sultana, S. K. Jung, and T. Bouwmans, "The emerging field of graph signal processing for moving object segmentation," in *Proc. Int. Workshop Frontiers Comput. Vis.* Springer, 2021, pp. 31–45.

[11] Y. Feng, Z. Feng, W. Hua, and Y. Sun, "Multimodal-XAD: Explainable autonomous driving based on multimodal environment descriptions," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 12, pp. 19469–19481, Dec. 2024.

[12] Y. Feng and Y. Sun, "Polarpoint-bev: Bird-eye-view perception in polar points for explainable end-to-end autonomous driving," *IEEE Trans. Intell. Vehicles*, early access, 2024, doi: 10.1109/TIV.2024.3361093.

[13] Y. Sun, W. Zuo, H. Huang, P. Cai, and M. Liu, "PointMoSeg: Sparse tensor-based end-to-end moving-obstacle segmentation in 3-D LiDAR point clouds for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 510–517, Apr. 2021.

[14] X. Chen et al., "Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6529–6536, Oct. 2021.

[15] S. Andreas Baur, D. Josef Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger, "SLIM: Self-supervised LiDAR scene flow and motion segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13126–13136.

[16] Z. Rozsa, A. Madaras, and T. Sziranyi, "Efficient moving object segmentation in LiDAR point clouds using minimal number of sweeps," *IEEE Open J. Signal Process.*, vol. 6, pp. 118–128, 2025.

[17] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 194–210.

[18] Y. B. Can, A. Liniger, D. P. Paudel, and L. Van Gool, "Structured bird's-eye-view traffic scene understanding from onboard images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 15661–15670.

[19] S. Gao, Q. Wang, and Y. Sun, "Obstacle-sensitive semantic bird-eye-view map generation with boundary-aware loss for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2024, pp. 466–471.

[20] S. Gao, Q. Wang, D. Navarro-Alarcon, and Y. Sun, "Forecasting semantic bird-eye-view maps for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2024, pp. 509–514.

[21] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 445–452, Apr. 2019.

[22] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4867–4873, Jul. 2020.

[23] 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 Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11974–11981, Oct. 2022.

[24] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[25] J. Houston et al., "One thousand and one hours: Self-driving motion prediction dataset," in *Proc. Conf. Robot Learn.*, Nov. 2021, pp. 409–418.

[26] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[28] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," in *Proc. ACM Multimedia Asia*, Dec. 2019, pp. 1–6.

[29] R. Gao, "Rethinking dilated convolution for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 4675–4684.

[30] X. Xiao et al., "BASeg: Boundary aware semantic segmentation for autonomous driving," *Neural Netw.*, vol. 157, pp. 460–470, Jan. 2023.

[31] Y. Wang, H. K. Chu, and Y. Sun, "PEAFusion: Parameter-efficient adaptation for RGB-thermal fusion-based semantic segmentation," *Inf. Fusion*, vol. 120, Aug. 2025, Art. no. 103030.

[32] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.

[33] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robot. Auto. Syst.*, vol. 108, pp. 115–128, Oct. 2018.

[34] J. Vertens, A. Valada, and W. Burgard, "SMSnet: Semantic motion segmentation using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 582–589.

[35] S. D. Jain, B. Xiong, and K. Grauman, "FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3664–3673.

[36] Y. Liu and H. Wang, "MotionRFCN: Motion segmentation using consecutive dense depth maps," in *Proc. Pacific Rim Int. Conf. Artif. Intell.* Springer, 2019, pp. 510–522.

[37] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, "MODNet: Motion and appearance based moving object detection network for autonomous driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2859–2864.

[38] X. Chen et al., "Automatic labeling to generate training data for online LiDAR-based moving object segmentation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6107–6114, Jul. 2022.

[39] J. Sun et al., "Efficient spatial–temporal information fusion for LiDAR-based 3D moving object segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 11456–11463.

[40] J. Kim, J. Woo, and S. Im, "RVMOS: Range-view moving object segmentation leveraged by semantic and motion features," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8044–8051, Jul. 2022.

[41] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biol. Cybern.*, vol. 64, no. 3, pp. 177–185, Jan. 1991.

[42] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 11138–11147.

[43] I. Dwivedi, S. Malla, Y.-T. Chen, and B. Dariush, "Bird's eye view segmentation using lifted 2D semantic features," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2021, pp. 6985–6994.

[44] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13760–13769.

[45] A. Hu et al., "FIERY: Future instance prediction in bird's-eye view from surround monocular cameras," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15273–15282.

[46] Z. Liu et al., "Vision-based uneven bev representation learning with polar rasterization and surface estimation," in *Proc. Conf. Robot Learn.*, 2023, pp. 437–446.

[47] Z. Li et al., "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 1–18.

[48] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17864–17875.

[49] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[50] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[51] Y. Li et al., "BEVDepth: Acquisition of reliable depth for multi-view 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 2, pp. 1477–1485.

[52] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[53] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.

[54] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[55] H. Rashed, M. Essam, M. Mohamed, A. Ei Sallab, and S. Yogamani, "BEV-MODNet: Monocular camera based bird's eye view moving object detection for autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1503–1508.

[56] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

Shiyu Meng (Student Member, IEEE) received the B.S. degree from Shaanxi University of Science and Technology, Xi'an, China, in 2017, and the M.S. degree in computer application technology from Northeast Forestry University, Harbin, China, in 2020. She is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong.

Her current research interests include moving-obstacle segmentation, computer vision, autonomous driving, and deep learning.

Yuxiang Sun (Member, IEEE) received the bachelor's degree from Hefei University of Technology, Hefei, China, in 2009, the master's degree from the University of Science and Technology of China, Hefei, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2017.

He is currently an Assistant Professor with the Department of Mechanical Engineering, City University of Hong Kong, Hong Kong. His current research interests include robotics and AI, autonomous driving, mobile robots, and autonomous navigation.

Prof. Sun serves as an Associate Editor for IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Intelligent Vehicles, IEEE Robotics and Automation Letters, IEEE International Conference on Robotics and Automation, and IEEE/RSJ International Conference on Intelligent Robots and Systems.