Supplementary Material for

MarS3D: A Plug-and-Play Motion-Aware Model for Semantic Segmentation on Multi-Scan 3D Point Clouds

1. Details of Methodology

In this section, we expound upon the proposed methodology by offering additional details and elucidations, thereby facilitating comprehension for our readership. To begin with, we denote all notations employed throughout the paper in Table 1.

1.1. Embedding in CFFE Module

In the 3D branch, the Cross-Frame Feature Embedding (CFFE) module introduces timestamp embeddings to bolster the model's temporal awareness for 3D representation learning. Specifically, as shown in Figure 1, temporal embeddings ($\mathcal{E}$) are summed up to the point-wise features (on the top left on Figure 1) of our 3D branch to indicate their respective timestamps. In other words, each point-wise feature is assigned with a distinct temporal embedding of $\mathcal{E}$ to indicate in which timestamp it is collected. $\mathcal{E}$ is randomly initialized at the beginning, and subsequently optimized alongside the other modules throughout network training.

1.2. Feature Fusion

Upon receiving multi-scan point clouds as input, the BEV branch and 3D branch respectively predict a multi-channel motion-aware feature map $\mathcal{Z}^m$ and enhanced spatial features $\mathcal{P}^s$. These two features are subsequently fused to achieve the ultimate prediction. As shown in Figure 2, for each feature of point in $\mathcal{P}^s \in \mathbb{R}^{N \times D_p}$ denoted as $p_i^s \in \mathbb{R}^{1 \times D_p}$, we query its corresponding BEV feature in $\mathcal{Z}^m \in \mathbb{R}^{H \times W \times D_z}$ according to its coordinate (x_i, y_i, z), and the dimension of the related feature is $1 \times D_z$. The two features from $\mathcal{P}^s$ and $\mathcal{Z}^m$ are then channel-wisely concatenated to derive the fused feature of the point as $\mathcal{P}_i^f \in \mathbb{R}^{1 \times (D_p + D_z)}$. The ultimate fused features $\mathcal{P}^f \in \mathbb{R}^{N \times (D_p + D_z)}$ is subsequently derived.

1.3. Prediction Head

The prediction head f_{cls} is devised to predict the final result. Figure 3 depicts the detailed architecture of the head. Since MarS3D is designed to predict both the semantic category of the input and its motion state, we design a dual-branch structure on the prediction head. The semantic branch is designed for the purpose of semantic prediction and the motion branch is intended to predict the motion states. During inference, the points recognized as movable categories are further predicted as either moving or static based on the prediction generated by the motion state branch.

Point-level Variables	
multi-scan point features	$\mathcal{P}^{\text{in}}$
temporal embeddings	$\mathcal{E}$
embedded features	$\mathcal{P}^{\text{ebd}}$
enhanced spatial features	$\mathcal{P}^s$
fused features	$\mathcal{P}^f$
BEV 2D Variables	
multi-scan BEV maps	$\mathcal{B}^{\text{in}}$
multi-scan feature maps	$\mathcal{U}$
discrepant feature maps	$\mathcal{D}$
motion-aware feature map	$\mathcal{Z}^m$
Networks	
2D CNN	f_u
multi-kernel CNN	f_m
NN layer	f_e
single-scan backbone	f_s
prediction head	f_{cls}
Logits	
semantic category logits	s^c_{pred}
motion logits	s^m_{pred}
Losses	
semantic category loss	L_c
motion loss	L_m

Table 1. Notations used in the paper.

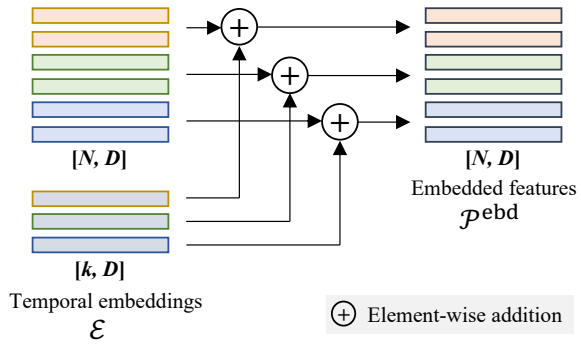

Figure 1. Visualization of the CFFE module. The top-left indicates features from the NN layer, and bottom-left indicates the trainable timestamp embeddings.

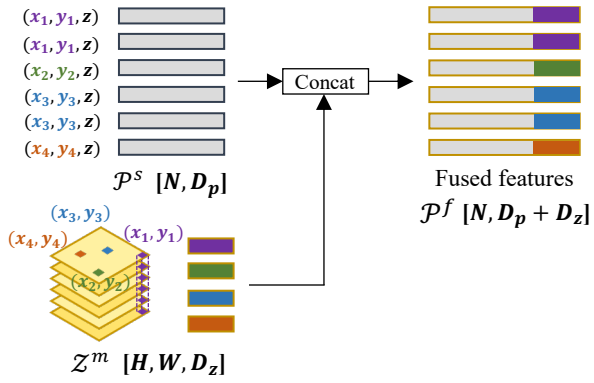

Figure 2. Illustration of feature fusion from BEV branch and 3D branch.

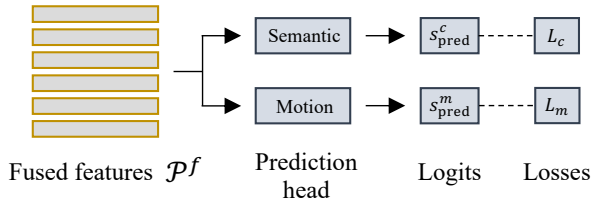

Figure 3. Dual-branch structure of the prediction head.

2. Compatibility of MarS3D

In order to show the compatibility of the proposed plug-and-play model, we adopt a variety of widely used models as backbones consisting of point-and-voxel-based approach SPVCNN [7] (implemented upon TorchSparse [6]), and voxel-based approaches MinkUNet (implemented upon Minkowski Engine [3]) and SparseConvUNet [5] (implemented upon SpConv [4]). Following with the track of SemanticKITTI, we choose models designed for single-scan tasks with only taking point clouds as input as the backbones, and these models are completely open source and highly reproducible.

SemanticKITTI [1] Multi-Scan Dataset	Ratio(%)
car (*car*)	4.08185
bicycle (*bic.*)	0.01661
motorcycle (*mot.*)	0.03984
truck (*tru.*)	0.20634
other-vehicle (*ove.*)	0.16497
person (*per.*)	0.01770
bicyclist (*bil.*)	1.11e-06
motorcyclist (*mol.*)	5.53e-07
road (*roa.*)	19.87965
parking (*par.*)	1.47172
sidewalk (*sid.*)	14.39230
other-ground (*ogr.*)	0.39049
building (*bui.*)	13.26862
fence (*fen.*)	7.23592
vegetation (*veg.*)	26.68150
trunk (*trn.*)	0.60350
terrain (*ter.*)	7.81422
pole (*pol.*)	0.28555
traffic-sign (*tra.*)	0.06156
moving-car (*mca.*)	0.17893
moving-bicyclist (*mbi.*)	0.01271
moving-person (*mpe.*)	0.01606
moving-motorcyclist (*mmo.*)	0.00375
moving-other-vehicle (*mov.*)	0.01574
moving-truck (*mtr.*)	0.01016
unlabeled	3.15018

Table 2. The ratio of the point number of each category (shown with full and short name) in SemanticKITTI [1] multi-scan dataset.

3. Details of Dataset

3.1. SemanticKITTI

The paper presents the performance of MarS3D on each category of SemanticKITTI [1], and the names of each category are abbreviated for the sake of brevity in the table. In Table 2, the short name and full name of each category are mapped. In addition, as shown in Table 2, the distribution of points across each category of SemanticKITTI is extremely imbalanced. Such a skewness can potentially hinder the performance of the model on certain categories, thereby emphasizing the need for future investigations to address this issue.

3.2. NuScenes

Based on the annotations of the 'lidar-seg' task and 3D object detection task of nuScenes [2] on the key frames of LiDAR point clouds, we designed a multi-scan setting and generated multi-scan segmentation annotations for all key frames. This label expands the 16 categories of single-scan 'lidar-seg' task to 24 categories, distinguishing the

Method	Backbone	mIoU	car			bicyclist			person			motorcyclist			other-vehicle			truck		
			m.	*n.*	*a.*	*m.*	*n.*	*a.*	*m.*	*n.*	*a.*	*m.*	*n.*	*a.*	*m.*	*n.*	*a.*	*m.*	*n.*	*a.*
Baseline	SPVCNN [7]	49.7	74.3	93.9	84.1	86.7	0.0	43.4	55.0	19.7	37.4	0.0	0.0	0.0	0.0	33.0	16.5	0.0	68.0	34.0
Ours		54.7	80.6	95.6	88.1	94.9	0.0	47.4	68.0	27.9	48.0	0.0	0.0	0.0	3.6	51.5	27.6	0.0	79.4	39.7
Baseline	SparseConv [5]	49.0	73.9	94.7	84.3	85.4	0.2	42.8	53.6	17.3	35.4	0.0	0.0	0.0	0.0	43.4	21.7	0.0	69.6	34.8
Ours		54.6	83.5	96.6	90.0	94.4	0.0	47.2	68.9	26.9	47.9	0.0	0.0	0.0	0.0	64.8	32.4	0.0	83.3	41.6
Baseline	MinkUNet [3]	48.5	69.2	93.8	81.5	83.1	0.0	41.6	52.5	18.0	35.2	0.0	0.0	0.0	0.0	41.3	20.6	0.0	90.3	45.2
Ours		54.7	82.6	96.4	89.5	93.1	0.0	46.6	64.4	31.6	48.0	0.0	0.0	0.0	0.1	62.7	31.4	0.0	93.9	47.0

Table 3. The quantitative results on SemanticKITTI [1] validation set. There are six categories that have two motion states: moving (indicated as *m.*) and non-moving (indicated as *n.*). Also, we calculate the average mIoU (indicated as *a.*) for each semantic category for a more comprehensive evaluation.

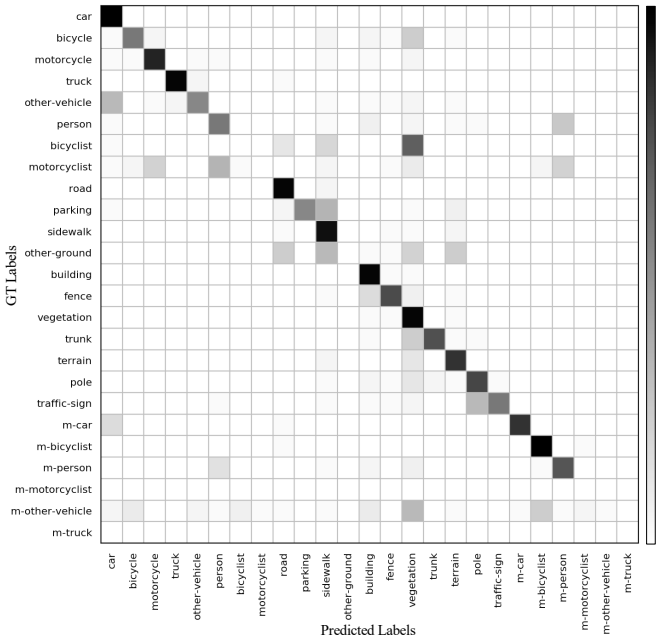

Figure 4. Confusion matrix of the model with SPVCNN as the backbone. Darker colors mean higher True Positive values.

motion state of categories that may move. Under this setting, non-key frames without annotations serve as reference frames fused with key frames as the input of the model, and the training is completed using supervision from only key frames, but not non-key frames, which we call semi-sequential supervised learning. In this case, according to the supervised learning paradigm, some voxels containing excessive unlabeled points may lead to confusion during training, thus affecting the performance of the model. In order to avoid being affected by the lack of the supervision when evaluating the performance improvement brought by our proposed plug-and-play module, instead of using each point as an evaluation element to calculate mIoU, we can also consider each voxel as an evaluation element for this task to calculate mIoU.

4. Additional Results and Discussion

In SemanticKITTI [1], there are six semantic categories that have two motion states, and we conduct statistics on the performance of MarS3D on these categories, as observed in Table 3. To better understand and analyze the results, we further provide the confusion matrix in Figure 4 based on SPVCNN [7] backbone. Besides, we provide more visualizations of different feature maps in Figure 5. MarS3D has improved performance in almost every category. Especially, the model demonstrates high accuracy in recognizing the motion state of objects.

For the evaluation results on nuScenes [2], We attempt to evaluate the trained model at both the voxel level and the point level in the absence of non-key frame (reference frame) supervision. We find that the performance depends on the semantic perception ability of different backbones that can deal with the information of unlabeled point clouds in shape and structure. This direction is worth exploring.

We run the main experiments and ablation studies multiple times, and obtain statistical results as shown in Table 4. The results indicate that MarS3D exhibits a robust perception and motion awareness capability in most semantic categories. In conclusion, MarS3D achieves significant improvements over all baseline methods, including the state-of-the-art. Additionally, MarS3D demonstrates robust motion perception, which can enhance the safety and intelligence of autonomous driving systems.

The proposed model presented in this study demonstrates the ability to classify both semantic categories and motion states. For independent evaluations of them, Table 5 demonstrates the substantial improvement brought by MarS3D. Furthermore, our findings indicate that the improvement in semantic perception stems from the proposed model itself rather than the incorporation of additional labels. To support this claim, we conducted experiments where we trained a model without motion state supervision based on SPVCNN. The results indicated that the model

Figure 5. More zoom-in discrepant feature maps from the inference of MarS3D are shown compared with motion BEV ground truth. Brighter region means higher confidence of moving objects. Our prediction is highly consistent with the GT.

Backbone	*baseline*	+CFFE	+BEV	+CFFE+BEV	MarS3D
SPVCNN	49.7±0.1	50.6±0.4	53.3±0.1	53.9±0.1	54.7±0.1
SparseConv	48.8±0.2	51.3±0.2	53.9±0.1	54.2±0.2	54.5±0.1
MinkUNet	48.1±0.4	52.1±0.1	53.1±0.3	54.1±0.1	54.6±0.1

Table 4. Main results and ablation studies on SemanticKITTI [1] public validation set.

Method	Backbone	semantic labels	motion states
baseline	SPVCNN [7]	60.8±0.2	86.4±0.1
Ours		66.5±0.1	90.2±0.1
baseline	SparseConv [5]	59.9±0.3	86.5±0.1
Ours		66.2±0.2	91.2±0.1
baseline	MinkUNet [3]	59.4±0.2	85.4±0.3
Ours		66.4±0.1	90.1±0.1

Table 5. Results for semantic labels and motion states on public validation set of SemanticKITTI.

attained a mean Intersection over Union (mIoU) score of 66.0% on semantic categories.

In this study, our main results are based on the analysis of three point cloud frames. To investigate the impact of temporal information, we also explore different frame numbers, including training without any temporal information (*i.e.*, single frame). Our findings, presented in Table 6, consistently demonstrate that models trained with multiple frames outperform those trained with only a single frame. Moreover, our results also indicate that the performance of models trained with multiple frames increases with the number of frames used for training.

Number of frames	1	2	3
mIoU(%)	51.7±0.3	52.8±0.1	54.7±0.1

Table 6. Ablation studies on the number of frames.

References

[1] 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 *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 2, 3, 4

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 3

[3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2, 3, 4

[4] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022. 2

[5] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 2, 3, 4

[6] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. TorchSparse: Efficient Point Cloud Inference Engine. In *Conference on Machine Learning and Systems (MLSys)*, 2022. 2

[7] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European conference on computer vision*, pages 685–702. Springer, 2020. 2, 3, 4