BE-STI: Spatial-Temporal Integrated Network for Class-agnostic Motion Prediction with Bidirectional Enhancement

Yunlong Wang[1,2*†], Hongyu Pan[1†], Jun Zhu[1], Yu-Huan Wu[1,3*], Xin Zhan[1], Kun Jiang[2‡], Diange Yang[2‡]
[1]Alibaba DAMO Academy, [2]Tsinghua University, [3]Nankai University

Abstract

Determining the motion behavior of inexhaustible categories of traffic participants is critical for autonomous driving. In recent years, there has been a rising concern in performing class-agnostic motion prediction directly from the captured sensor data, like LiDAR point clouds or the combination of point clouds and images. Current motion prediction frameworks tend to perform joint semantic segmentation and motion prediction and face the trade-off between the performance of these two tasks. In this paper, we propose a novel Spatial-Temporal Integrated network with Bidirectional Enhancement, BE-STI, to improve the temporal motion prediction performance by spatial semantic features, which points out an efficient way to combine semantic segmentation and motion prediction. Specifically, we propose to enhance the spatial features of each individual point cloud with the similarity among temporal neighboring frames and enhance the global temporal features with the spatial difference among non-adjacent frames in a coarse-to-fine fashion. Extensive experiments on nuScenes and Waymo Open Dataset show that our proposed framework outperforms all state-of-the-art LiDAR-based and RGB+LiDAR-based methods with remarkable margins by using only point clouds as input.[1]

1. Introduction

Modern self-driving vehicles are expected to operate in open traffic scenes with highly dynamical moving objects instead of only in closed scenes [1, 19, 32]. The motion of inexhaustible categories of traffic participants is critical for the safety of autonomous driving systems.

Traditional methods tend to formulate this task as trajectory prediction [1–5, 11, 14, 42, 44, 45], which lacks the ability to handle unexpected categories that have not been seen in training set due to the dependence on a separate object detector [27–29, 40, 41]. Scene flow [7, 8, 15, 18, 20, 33, 36] provides an attractive solution by estimating the dense motion field directly from LiDAR point clouds, which is computationally prohibitive for practical self-driving systems [17, 35]. Recent works seek to perform joint semantic segmentation and motion prediction based on BEV occupancy grids, which essentially solves a joint optimization problem and thus faces the trade-off between the performances of the above two tasks. Our goal in this paper is to explore a better way to combine the two tasks of semantics and motion, which utilizes a bidirectional enhancement network to improve the motion prediction performance through more accurate spatial semantic features.

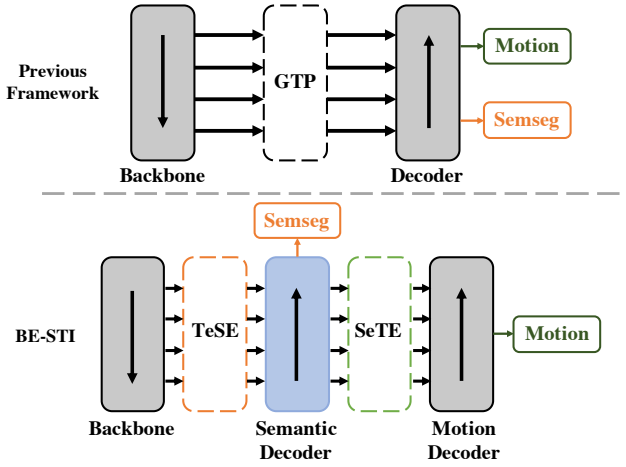

Figure 1. Comparison between BE-STI and previous motion prediction framework. **Top row:** Previous framework, which adopts global temporal pooling (GTP) to capture motion clues. **Bottom:** Our proposed BE-STI framework.

One may ask: whether semantic information can benefit motion prediction? We first seek to answer this question through a toy example. As shown in Tab. 1, we feed MotionNet [35], previous state-of-the-art (SOTA) LiDAR-based motion prediction framework, with semantic ground truth (GT) as additional input, which outperforms the previous work by a great margin. Specifically, we first use the segmentation GT of the points in the voxel to count the

*This work was done when Yunlong Wang and Yu-Huan Wu were research interns at Alibaba DAMO Academy.
†Equal contribution.
‡Corresponding author.
[1]The code will be released at https://github.com/be-sti/be-sti.

Method	Motion Prediction Mean Error (m) ↓		
	Static	Speed $\leq$ 5 m/s	Speed $>$ 5 m/s
MotionNet [35]	0.0201	0.2292	0.9454
MotionNet+ GT_{seg}	**0.0015**	**0.2139**	**0.7990**

Table 1. Performance on the motion prediction task on nuScenes.

distribution of the segmentation category as a segmentation vector, and then combine the vector with the original input to obtain a new input. Thus, we are confirmed high-quality semantic information have a positive impact on motion prediction task. To this end, we introduce a semantic decoder between our backbone network and motion decoder to obtain more accurate semantic information.

Considering that single frame of LiDAR information is sparse and the adjacent LiDAR frames describe similar scenes, the temporal information helps to extract more stable and accurate spatial semantic information. Given this, we propose a **temporal-enhanced spatial encoder (TeSE)** to perform better spatial understanding of each individual frame. TeSE is introduced to capture the common feature of neighboring frames and merge it to feature maps of each individual frame. In this way, the difficulty of individual spatial understanding caused by the sparsity of LiDAR points can be effectively compensated by adjacent frames.

Another concern about our framework is how to fully utilize the semantic information generated by our semantic decoder. To efficiently and effectively utilize the semantic feature, we propose a **spatial-enhanced temporal encoder (SeTE)** between the semantic decoder and motion decoder, which is designed to capture motion clues by discovering the spatial variation with time. Notice that temporal non-adjacent frames describe distinct scenes, we introduce SeTE to capture the discriminative feature of non-adjacent frames in time channel and feed it into motion decoder.

We name our novel proposed class-agnostic motion prediction network as **bidirectional enhanced spatial-temporal integrated network (BE-STI)**. We provide a comparison between the sketches of BE-STI and previous framework in Fig. 1. As can be seen, apart from the traditional backbone and motion decoder, we introduce three stacked modules: (1) TeSE, which is applied to perform spatial feature enhancement with temporal common features; (2) semantic decoder, which is applied to render motion prediction modules with auxiliary semantic information; (3) SeTE, which is applied to enhance the temporal motion features with spatial discriminative features.

The implementation of BE-STI is quite simple but efficient. All modules are built up with several stacked 2D and 3D convolution layers. Without any fancy structures, BE-STI surpasses all previous SOTA LiDAR-based and RGB+LiDAR-based methods on nuScenes dataset with only LiDAR point clouds input while running at over 22Hz.

Our contributions can be summarized as follows:

• We propose a novel class-agnostic motion prediction framework, named BE-STI, in which the benefits of semantic information to motion prediction is extensively explored. With the auxiliary semantic information in our framework, the motion prediction performance is significantly improved.

• We propose TeSE and SeTE to perform bidirectional enhancement between spatial and temporal feature extraction, in which TeSE contributes to the spatial understanding of each individual frame while SeTE captures high-quality motion clues by extracting spatial discriminative features.

• Extensive experiments on nuScenes and Waymo Open Dataset (WOD) demonstrates that BE-STI framework outperforms previous SOTA methods by a remarkable margin.

2. Related Work

2.1. Trajectory Prediction

Trajectory prediction aims to predict the future positions of some typical objects according to historical observations [3, 5, 11, 42, 44], typically involving three sub-modules: detection [12, 21, 22, 27–29, 39–41, 47], tracking [10, 26, 30, 34, 37, 38, 43, 46, 48] and prediction [1, 2, 4]. One of mainstream methods [5] is to predict the object trajectories in a cascading manner where the three sub-modules receive the output of previous sub-module, respectively. Considering the limited information sharing among these sub-modules, such strategy sacrifices the potential advantage of joint optimization. Another method [44] designs an end-to-end neural network to perform three tasks jointly and achieve great performance improvement. However, this approach has greater difficulty and longer epochs to optimize the network.

Compared with trajectory prediction, our proposed method pays more attention to capture of motion information and gets rid of the dependence on the detection results. When the detection is inaccurate or some moving objects, which are not seen in the training set, such as balls, small animals, our method still could provide accurate motion prediction, which improves the completeness of the system of self-driving vehicles.

2.2. Flow Estimation

This task aims to estimate flow to describe the motion from past to current time. 3D flow, which is also called scene flow [7–9, 15, 18, 20, 33, 36], is a hot research topic recently. The task is to estimate the motion from LiDAR points and attach each point with a 3D vector to represent the dense 3D motion field. Current scene flow methods are trained and tested on either densely processed data KITTI Scene Flow [7], or synthetic data FlyingThings3D [18]. However, it is difficult to directly apply these methods on

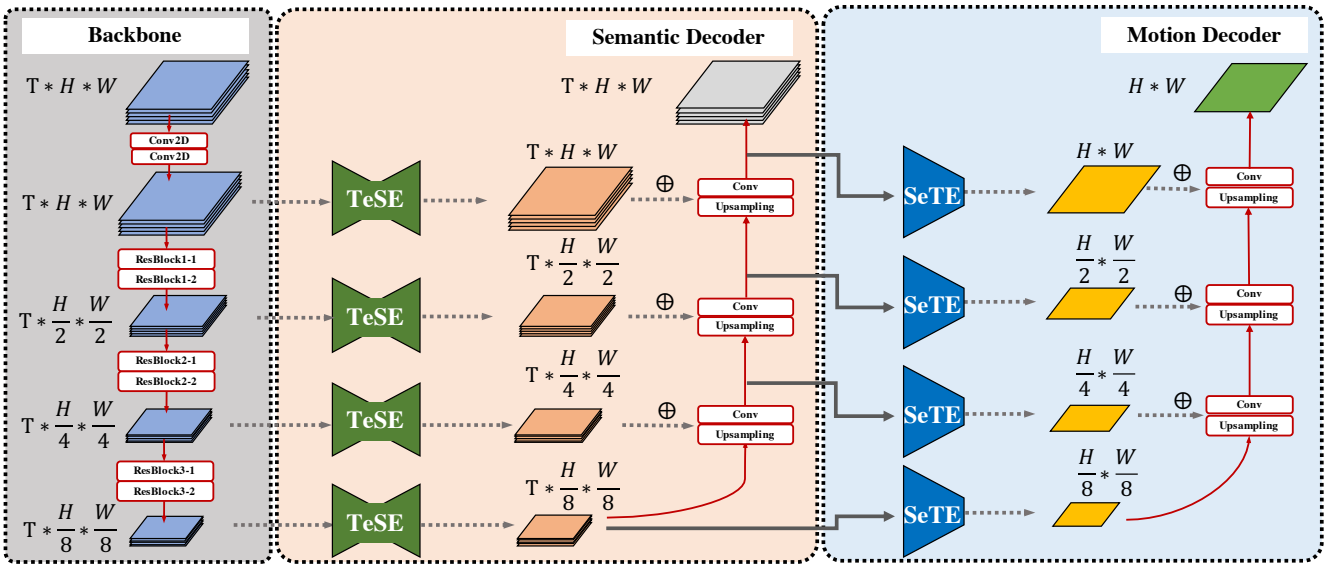

Figure 2. Architecture of spatial-temporal integrated network with bidirectional enhancement (**BE-STI**). **Left:** Backbone. **Middle:** Spatial semantic decoder stage. **Right:** Temporal motion decoder stage.

real point clouds obtained by LiDAR, which have no one-to-one correspondences between pairs of points in past and current point clouds.

Since self-driving system rarely cares about the vertical motion of surrounding agents, there is a growing trend towards estimating the motion and predicting the future position in BEV [6, 13, 17, 23, 35]. MotionNet [35] is the prior in this direction, which proposes to jointly perform perception and motion prediction on 2D BEV maps. PillarMotion [17] proposes a cross-sensor based self-supervision to train MotionNet [35] with the additional optical flow supervision from RGB images, which significantly improve the motion prediction accuracy by a great margin when combining the self-supervised model with supervised fine-tuning.

Although we believe that the two tasks have a mutually promoting relationship, the information required by the two tasks is quite different. So we think it is not suitable to use the same feature to solve the two tasks like MotionNet [35]. Our proposed method explore a better way to combine the two tasks, which first extracts a semantic feature with more accurate precision, and further uses this feature to obtain motion information. Through TeSE and SeTE to enhance the relationship of space and temporal, the network has the ability to extract better semantic and motion information.

3. The proposed Approach

3.1. Problem Formulation

Given a temporal sequence of LiDAR point clouds obtained by the moving self-driving vehicle, an ego-motion compensation module described in MotionNet [35] is applied first to synchronize all the past frames to the current coordinate system of ego vehicle. We denote each synchronized point cloud at time t as $P^t = \{P_i^t\}_{i=1}^{N_t}$, where $P_i^t \in R^3$ indicates the coordinate of a point and N_t is the number of points. Then P^t is discretized into dense 3D voxels $V^t \in \{0,1\}^{H\times W\times C}$, where the empty voxel is represented by 0, the non-empty one is represented by 1 and H, W, C are the numbers of voxels along X, Y and Z axis. After that, we represent V^t as a 2D pseudo-image with the vertical dimension corresponding to image channels, which can be regard as a BEV map with C channels. Therefore, the BEV motion field is defined as the movement of each gird to its corresponding position at next timestamp, which can be denoted as $M^t \in R^{H\times W\times 2}$. The movement of each point P_i^t is simply identified as the movement of the corresponding BEV grid.

3.2. Bi-enhanced Spatial-temporal Network

As mentioned in Sec. 1, the pre-experiment result of a toy example demonstrates the improvement of semantic information on motion prediction task. To this end, we propose to introduce a semantic decoder in our framework to boost the performance of motion prediction. Given the fact that the sparsity of LiDAR points makes it tough to perform high-quality spatial understanding while the temporal adjacent frames capture similar scenes, TeSE is introduced in our framework to perform better spatial feature extraction on each individual frame with the assistance of temporal global features. Besides, we also design a SeTE module to efficiently capture the temporal motion clues through the spatial discriminative features of non-adjacent frames. The overall structure is illustrated in Fig. 2.

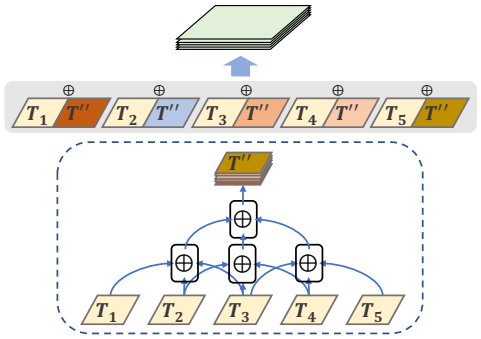

Figure 3. Temporal-enhanced Spatial Encoder (**TeSE**)

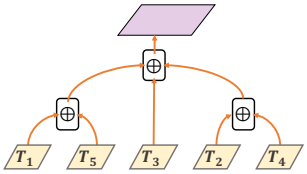

Figure 4. Spatial-enhanced Temporal Encoder (**SeTE**)

3.2.1 Temporal-enhanced Spatial Encoder

We are confirmed that the performance of motion prediction can be further improved by introducing better semantic features. Notice that the sparsity of the point cloud makes spatial understanding tough while the adjacent frames describe similar scenes, the temporal common features can benefit the spatial feature encoding of each individual frame. Therefore, we introduce TeSE here to extract common features of adjacent frames and feed back to the feature map of each individual frame.

TeSE consists of two stages: global temporal feature extraction and individual-global feature fusion, see Fig. 3. The global temporal feature extraction is built up with several stacked 3D convolution layers with kernel size of $k \times 3 \times 3$, where k corresponds to the temporal dimension. During this stage, the temporal dimension is gradually reduced to 1 and the channel increases to T times, which corresponds to the global feature representation of the T frames, respectively. Practically, we apply a 2D convolution layer with kernel size 3×3 and a 3D convolution layer with kernel size $k \times 1 \times 1$ to replace the $k \times 3 \times 3$ convolution for the purpose of model complexity reduction. During the individual-global feature fusion stage, the extracted global temporal feature is divided into T parts along the channel and stacked with the feature map of each individual pseudo-image in the sequence. Then a 3D convolution layer with kernel size of $2 \times 3 \times 3$ is applied to each pair of individual-global feature map for the purpose of enhanced spatial feature encoding.

3.2.2 Spatial-enhanced Temporal Encoder

Notice that the motion clues of objects are implicit in the changes of the scene, which is mainly represented by the discriminative features among non-adjacent frames. SeTE is introduced here to capture high-quality motion information from non-adjacent frames by discovering the spatial variation with time.

Previous works usually take global temporal pooling (GTP) to capture the temporal features, which pays equal attention to all temporal frames. We notice that the combination between two frames with different temporal intervals can provide distinct motion clues. Specifically, according to the comparison between the first frame and the last frame in a temporal sequence, it is easy to coarsely determine the moving speed and the motion state of each object. Then the trajectory clues provided by the temporal intermediate frames can be applied to model fine motions.

To this end, given a sequence of feature maps which encode the spatial features of each frame. Here we take $T = 5$ frames as example. The order among frames in the sequence is organized as shown in Fig. 4. In SeTE, a convolution layer with kernel size $2 \times 3 \times 3$ is first applied to capture the feature of the first and the last frame, which is also applied to capture the feature of the second and the penultimate frame, etc. Then the generated two feature maps together with the middle frame are fed into a $3 \times 3 \times 3$ convolution to capture the global motion feature.

3.2.3 Architecture of BE-STI framework

We propose to fully utilize the semantic information to boost the motion prediction performance in our BE-STI framework. Apart from (1) how to introduce semantic feature extraction during the design of motion prediction network, there still exists another two issues to be addressed: (2) how to perform better spatial feature extraction for each individual pseudo-image, which is essential for better semantic understanding; (3) how to aggregate temporal features based on spatial semantic feature representation of each individual pseudo-image.

To address the above issues, except for the traditional backbone network and motion decoder, we introduced additional three modules, TeSE, semantic decoder and SeTE, in BE-STI framework. As shown in Fig. 2, we intuitively divide the whole framework into two stages, which are spatial semantic stage and temporal motion decoder stage.

Spatial semantic decoder stage. Given a sequence of pseudo-images, four stacked blocks are applied as the backbone to extract multi-scale feature maps of each individual image. Except for the first block, which is composed with two 2D convolutions of stride 1, the other three blocks are built up with classical 2D ResBlocks. Each block downsamples the feature map by half on spatial dimension via the first layer which contains a convolution with a stride of size 2. Notice that each pseudo-image is operated individually and there is no downsampling or any other operation

on temporal dimension. For the sequence of feature maps at each spatial scale, TeSE is applied to enhance the spatial feature representation of each individual pseudo-image with temporal common features. After that, a bottom-up semantic decoder is applied to merge the feature maps of each image at different spatial scales, which consists of stacked upsampling blocks. Each block is built up with stacked upsampling, concatenate and convolution layers. The output of the semantic decoder is a sequence of feature maps with the same size of the input pseudo-images, where each feature map corresponds to a pseudo-image. Here we use semantic segmentation task to supervise the feature learning during the spatial semantic stage.

Temporal motion decoder stage. The input to the temporal motion decoder stage is the multi-scale feature maps of each individual pseudo-image, which is generated by the upsampling blocks of the semantic decoder. For the sequence of feature maps at each spatial scale, SeTE is applied to capture the spatial discriminative features. After that, the spatial discriminative features at multi-scale are delivered to the bottom-up motion decoder, which consists of the same structure with the semantic decoder. The output of motion decoder is then fed into the same heads as described in MotionNet: (1) motion-prediction head, which predicts the future position of each grid cell in the BEV pseudo-image; (2) state-estimation head, which predicts whether a cell is static or moving; (3) cell-classification head, which predicts the semantic of each cell. Notice that the cell-classification result is only used to classify between foreground cells and background cells, which is applied to suppress the jitters of background cells. The motion prediction head do not rely on the cell-classification result and is able to predict the motion of unseen objects beyond training set.

3.2.4 Loss Function

Our proposed BE-STI network is trained with the assistance of four losses associated with four heads, which are (1) the motion prediction loss, state estimation loss and cell classification loss after the motion decoder; (2) the semantic segmentation loss of all pseudo-images after the spatial encoder.

Motion prediction loss. We apply weighted smooth L1 loss for the motion prediction head, where each category is assigned with a weight to balance the amount of grid cells in different categories:

$$\mathcal{L}_{\text{mot}} = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot \text{SmoothL1}(x_{\text{mot},i}, x^{\text{gt}}_{\text{mot},i}) \qquad (1)$$

where $\text{SmoothL1}(\cdot)$ is smooth L1 loss, N is the number of non-empty grid cells in current pseudo-image, $x_{\text{mot},i}$ is the predicted displacement of each non-empty cell, $x^{\text{gt}}_{\text{mot},i}$ is the corresponding ground truth and w_i is the weight assigned to different categories:

$$w_i = \begin{cases} 0.005, & i\text{-th cell} \in \text{background} \\ 1.0 \quad , & \text{else} \end{cases} \qquad (2)$$

Motion state estimation loss. We adopt cross-entropy loss for the motion state estimation head:

$$\mathcal{L}_{\text{state}} = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot \text{CE}(x_{\text{state},i}, x^{\text{gt}}_{\text{state},i}) \qquad (3)$$

where $\text{CE}(\cdot)$ is cross-entropy loss, N is the number of non-empty grid cells in current pseudo-image, $x_{\text{state},i}$ is the predicted motion state of each non-empty cell, $x^{\text{gt}}_{\text{state},i}$ is the corresponding ground truth and w_i is the weight assigned to the cell categories defined in Eq. (2).

Cell classification loss. We also apply cross-entropy loss for the cell classification head:

$$\mathcal{L}_{\text{cls}} = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot \text{CE}(x_{\text{cls},i}, x^{\text{gt}}_{\text{cls},i}) \qquad (4)$$

where $x_{\text{cls},i}$ is the predicted category of each non-empty cell, $x^{\text{gt}}_{\text{cls},i}$ is the corresponding ground truth. $\text{CE}(\cdot)$, N and w_i are previously defined in motion state estimation loss.

Semantic segmentation loss The cross-entropy loss is also adopted for the assistant semantic segmentation task:

$$\mathcal{L}_{\text{seg}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t} w_i \cdot \text{CE}(x^{t}_{\text{seg},i}, x^{t,\text{gt}}_{\text{seg},i}) \qquad (5)$$

where $\text{CE}(\cdot)$ is cross-entropy loss, T denotes the number of pseudo-images, N_t is the number of non-empty grid cells in the t-th image, $x^{t}_{\text{seg},i}$ is the predicted category of each non-empty cell, $x^{t,\text{gt}}_{\text{seg},i}$ is the corresponding ground truth, w_i is previously defined in Eq.(2).

Therefore, the total loss for the training of BE-STI is defined as follows:

$$\mathcal{L} = \lambda_{\text{mot}} * \mathcal{L}_{\text{mot}} + \lambda_{\text{state}} * \mathcal{L}_{\text{state}} + \lambda_{\text{cls}} * \mathcal{L}_{\text{cls}} + \lambda_{\text{seg}} * \mathcal{L}_{\text{seg}} \qquad (6)$$

where λ_{mot}, λ_{state}, λ_{cls} and λ_{seg} are the balancing factors to adjust the importance among four sub-tasks.

4. Experiments

The performance of BE-STI is evaluated on the challenging BEV motion prediction benchmark of nuScenes dataset. We first introduce the experimental setup in Sec. 4.1. In Sec. 4.2, we report main results including the comparison with SOTA methods, runtime analysis and qualitative results. Finally, we conduct extensive ablation studies to analyze BE-STI network in Sec. 4.3, including the implementation of MotionNet and BE-STI on WOD [31].

4.1. Experimental Setup

Dataset. All experiments are conducted on a large-scale autonomous driving dataset, nuScenes, which provides full

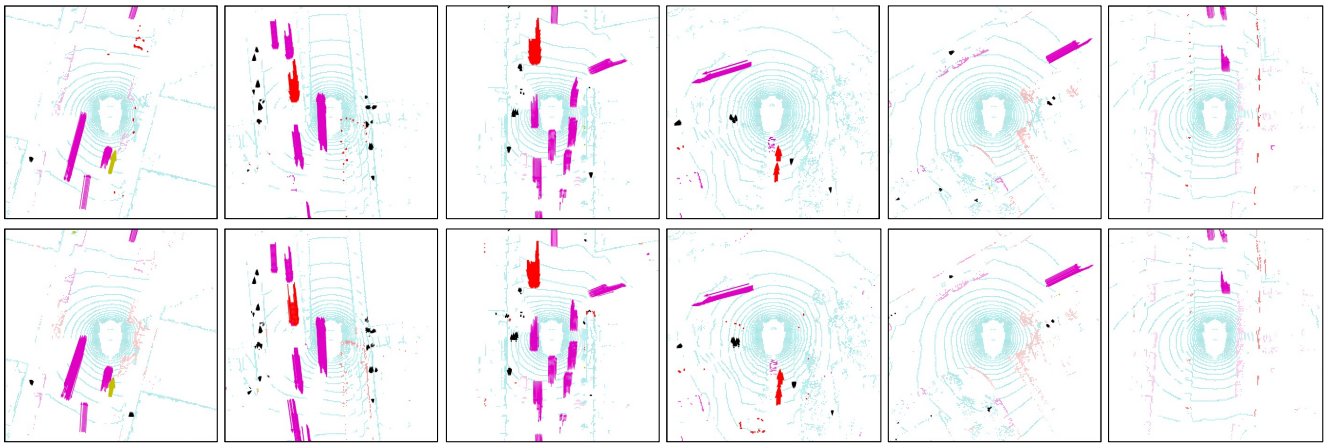

Figure 5. Qualitative results of the proposed BE-STI framework. Top row: ground truth. Bottom row: BE-STI predictions. We represent the motions with an arrow attached to each grid. The cell classification result is represented by various colors. Cyan: background; pink: vehicle; black: pedestrian; yellow: bike; red: others.

autonomous vehicle sensor suite, including 1 LiDAR, 6 cameras, 5 radars, GPS and IMU, all with full 360 degree coverage on the surroundings. nuScenes is composed with a total of 1000 scenes, in which 150 scenes are used as testing set and the annotations of them cannot be accessed. Therefore, we have 850 scenes with ground truth annotations in total, among which 500 scenes are used for training, 100 scenes for validation and and 250 scenes for testing following the tradition in previous works [17, 35]. For each scene, we only utilize the LiDAR point clouds during the training and testing of the network. The LiDAR point clouds are captured with a frequency of 20Hz and annotated with a frequency of 2 Hz. Besides, the sequences of LiDAR point clouds lasts about 20s in each scene. As described in [35], the ground truth motion can be derived from the bounding box annotations which is originally provided for detection and tracking tasks.

Implementation details. For fair comparison, we follow the same data pre-processing settings adopted by previous works [17, 35], where the input point clouds are cropped in the range of [-32m, 32m] $\times$ [-32m, 32m] $\times$ [-3m, 2m] and the voxel size is set to 0.25m $\times$ 0.25m $\times$ 0.4m along XYZ axis, respectively. The sequence length of LiDAR point clouds is set to five, where the last one corresponds to the current time while the previous four frames are from the past time. The time interval between adjacent frames is 0.2s. For the training of ancillary semantic segmentation and cell classification tasks, five categories are defined as below: Background, Vehicle, Pedestrian, Bicycle and Others, where "Others" represents the unseen objects with various appearances and motion behaviours beyond the training data.

As shown in Fig. 2, the input of BE-STI network is a 4D tensor with size $5 \times 13 \times 256 \times 256$, where $5, 13, 256$ corresponds to the temporal, channel and spatial sizes, respectively. We first apply two-layer 2D convolutions to lift its channel size to 32. Then stacked ResBlocks are applied to decrease the spatial size to $128, 64, 32$ and lift the channel size to $64, 128, 256$ gradually. All sizes of the tensor keep unchanged while the temporal size is reduced from 5 to 1 through SeTE.

Our BE-STI framework is implemented in Pytorch and trained in two stages with the AdamW [16] optimizer. For nuScenes dataset, we train the entire network with batch size 16 for 70 epochs on 8 Tesla V100 GPUs. We set the initial learning rate to 0.002 and decay it by a factor of 0.5 at the 20, 40, 50 and 60-th epochs. We set $\lambda_{\text{mot}} = \lambda_{\text{state}} = 1.0$, $\lambda_{\text{cls}} = \lambda_{\text{seg}} = 2.0$ during the beginning 30 epochs and set $\lambda_{\text{mot}} = \lambda_{\text{state}} = 1.0$, $\lambda_{\text{cls}} = 2.0$, $\lambda_{\text{seg}} = 0.0$ during the last 40 epochs. The purpose is to first improve the spatial feature extraction ability of BE-STI and then make the model focus on the motion prediction task. We adopt multi-gradient descent algorithm (MGDA) [25] during the training of our best model and disable it when performing ablation studies.

Evaluation metrics. Following MotionNet [35], we divide the non-empty cells into three groups according to their speeds: static($\leq$ 0.2 m/s), slow($\leq$ 5 m/s) and fast($>$ 5 m/s). Then we report the mean and median prediction error on each group, which is the L_2 distances between the predicted displacements and the ground truth displacements 1s into future. Besides, we also report the performance on auxiliary cell classification tasks. Here we report the average accuracy over all non-empty cells and the average accuracy over all five categories, which are denoted as overall accuracy (OA) and mean category accuracy (MCA), respectively.

4.2. Main Results

Comparison with SOTA methods. We extensively compare our proposed BE-STI framework with a variety of published algorithms in Tab. 2. According to the

Method	Modality	Static		Speed ≤ 5 m / s		Speed > 5 m / s	
		Mean ↓	Median ↓	Mean ↓	Median ↓	Mean ↓	Median ↓
FlowNet3D [15]	L	0.0410	0	0.8183	0.1782	8.5261	8.0230
HPLFlowNet [8]	L	**0.0041**	0.0002	0.4458	0.0960	4.3206	2.4881
PointRCNN [28]	L	0.0204	0	0.5514	0.1627	3.9888	1.6252
LSTM-ED [24]	L	0.0358	0	0.3551	0.1044	1.5885	1.0003
MotionNet [35]	L	0.0201	0	0.2292	0.0952	0.9454	0.6180
PillarMotion [17]	I&L	0.0245	0	0.2286	0.0930	0.7784	**0.4685**
BE-STI (ours)	L	0.0220	**0**	**0.2115**	**0.0929**	**0.7511**	0.5413

Table 2. Comparison with SOTA results on nuScenes. We report the mean and medium errors on three groups, which are static grids, moving grids with speed ≤ 5 m/s and moving grids with speed > 5 m/s. **L:** LiDAR-based method. **I&L:** Image+LiDAR based method.

Method	Classification Accuracy (%) ↑						
	Bg	Vehicle	Ped.	Bike	Others	MCA	OA
PointRCNN [28]	**98.4**	78.7	44.1	11.9	44.0	55.4	**96.0**
LSTM-ED [24]	93.8	91.0	73.4	17.9	71.7	69.6	92.8
MotionNet [35]	97.0	90.7	77.7	19.7	66.3	70.3	95.8
BE-STI (ours)	94.6	**92.5**	**82.9**	**25.9**	**77.3**	**74.7**	93.8

Table 3. Performance on the auxiliary cell classification task on nuScenes.

modality of training data, all published methods can be categorized into LiDAR-only methods and RGB+LiDAR methods, where PillarMotion is the one trained with well-calibrated RGB images and LiDAR points. As can be seen, our method reports a novel SOTA result with reference to the mean prediction error on slow and fast moving objects compared with all previous works. Specifically, when compared to LiDAR-only methods, our BE-STI framework outperforms the previous SOTA method MotionNet [35] with a great margin of 0.1943m mean error and 0.0767m median error for the fast speed group, 0.0177m mean error and 0.0023m median error for the slow speed group. Even compared to the previous best RGB+LiDAR method PillarMotion [17], BE-STI still outperforms it with a margin of 0.0273m mean error for the fast speed group, 0.0171 m mean error for the slow speed group. We also report the auxiliary cell classification results of our proposed method in Tab. 3. As we can see, our method performs higher accuracy on all of movable objects and MCA. The experimental results show our proposed method is a better way to combine the segmentation and motion tasks.

Runtime analysis. For autonomous driving systems, the LiDAR point cloud processing time should be strictly no more than 100ms. At the inference stage, our whole model runs as 45ms on a single Tesla V100 GPU, where the point cloud transformation and voxelization use 10ms and the forward procedure of our model takes 35ms. Therefore, our BE-STI network has the potential of practical application on self-driving systems.

Qualitative results. We show the qualitative results of our BE-STI structure in Fig. 5, in which the bottom row lists our predicted results and the top row lists the corresponding ground truth. The predicted motion is represented by an arrow attached to each grid cell, whose length and direction represents the displacement 1s into future. As we can see, BE-STI produces high-quality motion prediction results on BEV grid cells.

4.3. Ablation Studies

We first evaluate the contributions of each individual component in BE-STI, see Tab. 4. For each experiment from (a) to (d), the model is built up with the listed modules together with the backbone and motion decoder mentioned in Sec. 3.2. Except the global temporal pooling (GTP) is proposed in MotionNet [35], SeTE, TeSE and Semantic Decoder are the individual components designed in our BE-STI framework. (a) the baseline method, which adopts GTP; (b) SeTE; (c) SeTE and Semantic Decoder; (d) our full model, which adopts TeSE, SeTE, and Semantic Decoder.

SeTE. Compare (b) with (a), we can see that introducing SeTE can significantly decrease the prediction errors on all three groups of cells, and increase the MCA for segmentation slightly. Experiments show that the spatial information is obviously helpful for extracting better temporal features. It is worth noting that we only replace GTP with SeTE, the performance has surpassed MotionNet [35].

Semantic Decoder. Compare (c) with (b), the MCA has significantly been improved, cause the semantic decoder combines low-level and high-level feature, which helps to obtain more detailed spatial features. Meanwhile the prediction errors on all three groups of cells have been decreased, which proves that high-quality semantic feature is conducive to the extraction of motion information.

TeSE. Compare (d) with (c), the MCA has been further reduced, cause TeSE utilizes temporal information to enhance the semantic features, alleviating the problem of sparse information in single frame. Not surprisingly, the prediction errors of moving objects also dropped further,

Method	GTP	TeSE	Sem. Decoder	SeTE	Static		Speed ≤ 5 m / s		Speed >5 m / s		MCA↑
					Mean ↓	Median ↓	Mean ↓	Median ↓	Mean ↓	Median ↓	
MotionNet [35]	✓				0.0256	0	0.2565	0.0962	1.0744	0.7332	70.3
(a)	✓				0.0250	0	0.3014	0.0971	1.6326	0.8889	69.8
(b)				✓	0.0249	0	0.2477	0.0959	1.0429	0.7203	70.6
(c)			✓	✓	**0.0224**	0	0.2391	**0.0949**	0.9376	0.6404	72.9
(d)		✓	✓	✓	0.0244	**0**	**0.2375**	0.0950	**0.9078**	**0.6262**	**74.8**

Table 4. Performance comparison of our models with different combinations of components on nuScenes. The models listed here are implemented without MGDA [25] for purely evaluation of components.

Supp.	Static		Speed ≤ 5 m/s		Speed >5 m / s	
	Mean ↓	Median ↓	Mean ↓	Median ↓	Mean ↓	Median ↓
(e)	0.0471	0.0039	**0.2080**	0.0928	0.7245	**0.5411**
(f)	0.0415	0	0.2093	0.0930	0.7245	**0.5411**
(g)	**0.0220**	0	0.2115	0.0929	0.7511	0.5413
(h)	0.0223	**0**	0.2103	**0.0927**	**0.7510**	0.5413

Table 5. Motion prediction error with different post-processing methods on nuScenes.

Method	Motion Prediction Mean Error (m) ↓		
	Static	Speed ≤ 5 m/s	Speed >5 m / s
MotionNet [35]	0.0248	0.2950	1.3663
BE-STI (ours)	**0.0244**	**0.2850**	**1.1594**

Table 6. Motion prediction mean error on Waymo Open Dataset.

Method	Classification Accuracy (%) ↑						
	Bg	Vehicle	Ped.	Bike	Others	MCA	OA
MotionNet [35]	**96.7**	99.0	86.2	56.2	66.9	81.0	**96.4**
BE-STI (ours)	95.3	**99.4**	**90.0**	**76.7**	**79.0**	**88.1**	95.1

Table 7. Classification accuracy on Waymo Open Dataset.

which once again proves that more accurate semantic information contributes more to the extraction of motion clues.

Post-processing methods. Tab. 5 shows the effects of different post-processing methods. We extensively evaluate several suppression methods, which are: (e) with no post-processing; (f) if the predicted motion is less than 0.2 m/s, the final decision on the motion of corresponding cells is zero; (g) if the predicted motion is less than 0.2 m/s or the cell is predicted to be static or belongs to background, the final decision on the motion of corresponding cells is zero; (h) if the cell is predicted to be static or belongs to background, the final decision on the motion of corresponding cells is zero. From Tab. 5, we could find that our proposed method hardly predicts a very small flow for fast moving objects according to the comparison of (f) with (e). In addition, compare (g) with (f), the state of motion and the segmentation are helpful in suppressing the jitters significantly. For fair comparison, we adopt (g), the one adopted by previous works [17, 35], for our final result.

Performance on Waymo Open Dataset. For further comparison, we implement MotionNet and BE-STI on WOD [31]. The implementation details are the same with that on nuScenes, except for the training epoch is set to 25. There are a total of 146534 generated samples after pre-processing, among which 142919 samples are used for training and 3615 samples for testing. For motion prediction task, as shown in Tab. 6, BE-STI outperforms MotionNet with a margin of 0.0004 m, 0.0100m, 0.2069m mean error on static, slow and fast groups separately. For semantic segmentation task, our BE-STI still performs higher classification accuracy on all movable objects and MCA.

5. Conclusion

We have presented BE-STI, a novel framework for class-agnostic motion prediction from LiDAR point clouds. Our framework involves a TeSE which performs spatial feature enhancement for each individual point cloud based on the similarity among temporal adjacent frames and a SeTE which performs global temporal feature enhancement based on the spatial difference among non-adjacent frames. Experimental results on nuScenes dataset demonstrate that our proposed BE-STI framework significantly improves the motion prediction performance and reports a novel SOTA result compared with previous published methods. Besides, our framework can run at realtime and therefore is compatible with practical autonomous driving systems. Furthermore, we have also explored the bidirectional enhancement between semantic segmentation and motion prediction, which provides a more efficient way for the combination of various perception tasks like segmentation, prediction, etc. We hope the findings would promote the development of a more complete self-driving system in open traffic scenes.

Acknowledgments This work is supported by National Natural Science Foundation of China (U1864203). It is also supported in part by the National Key Research and Development Program of China (Grant NO. 2020AAA0108104), and by Alibaba Innovative Research (AIR) Program and Alibaba Research Intern Program. It is also supported in part by National Natural Science Foundation of China (61903220, 52102464), and in part by research project of sharingvan (HT20082302).

References

[1] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 1, 2

[2] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 1, 2

[3] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 1, 2

[4] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneider. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2095–2104, 2020. 1, 2

[5] Liangji Fang, Qinhong Jiang, Jianping Shi, and Bolei Zhou. Tpnet: Trajectory proposal network for motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6797–6806, 2020. 1, 2

[6] Artem Filatov, Andrey Rykov, and Viacheslav Murashkin. Any motion detector: Learning class-agnostic scene dynamics from a sequence of lidar point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9498–9504. IEEE, 2020. 3

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1, 2

[8] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3254–3263, 2019. 1, 2, 7

[9] Philipp Jund, Chris Sweeney, Nichola Abdo, Zhifeng Chen, and Jonathon Shlens. Scalable scene flow from point clouds in the real world. *IEEE Robotics and Automation Letters*, 2021. 2

[10] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):140–153, 2018. 2

[11] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125. IEEE, 2018. 1, 2

[12] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 2

[13] Kuan-Hui Lee, Matthew Kliemann, Adrien Gaidon, Jie Li, Chao Fang, Sudeep Pillai, and Wolfram Burgard. Pillarflow: End-to-end birds-eye-view flow estimation for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2007–2013. IEEE, 2020. 3

[14] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2020. 1

[15] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019. 1, 2, 7

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[17] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Self-supervised pillar motion learning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2021. 1, 3, 6, 7, 8

[18] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 1, 2

[19] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021. 1

[20] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 527–544. Springer, 2020. 1, 2

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 2

[22] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[23] Marcel Schreiber, Vasileios Belagiannis, Claudius Gläser, and Klaus Dietmayer. Dynamic occupancy grid mapping with recurrent neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6717–6724. IEEE, 2021. 3

[24] Marcel Schreiber, Stefan Hoermann, and Klaus Dietmayer. Long-term occupancy grid prediction using recurrent neural

networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9299–9305. IEEE, 2019. 7

[25] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *arXiv preprint arXiv:1810.04650*, 2018. 6, 8

[26] Sarthak Sharma, Junaid Ahmed Ansari, J Krishna Murthy, and K Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3508–3515. IEEE, 2018. 2

[27] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1, 2

[28] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 1, 2, 7

[29] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1, 2

[30] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5620–5629, 2017. 2

[31] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 5, 8

[32] Dequan Wang, Coline Devin, Qi-Zhi Cai, Philipp Krähenbühl, and Trevor Darrell. Monocular plan view networks for autonomous driving. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2876–2883. IEEE, 2019. 1

[33] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6954–6963, 2021. 1, 2

[34] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *arXiv preprint arXiv:2008.08063*, 2020. 2

[35] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11385–11395, 2020. 1, 2, 3, 6, 7, 8

[36] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *European Conference on Computer Vision*, pages 88–107. Springer, 2020. 1, 2

[37] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015. 2

[38] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3988–3998, 2019. 2

[39] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2

[40] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 1, 2

[41] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021. 1, 2

[42] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019. 1, 2

[43] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2365–2374, 2019. 2

[44] Zhishuai Zhang, Jiyang Gao, Junhua Mao, Yukai Liu, Dragomir Anguelov, and Congcong Li. Stinet: Spatio-temporal-interactive network for pedestrian detection and trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11346–11355, 2020. 1, 2

[45] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. *arXiv preprint arXiv:2008.08294*, 2020. 1

[46] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020. 2

[47] Yin Zhou and Oncel 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*, pages 4490–4499, 2018. 2

[48] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018. 2