Weakly Supervised Class-agnostic Motion Prediction for Autonomous Driving

Ruibo Li[1,2], Hanyu Shi[2], Ziang Fu[3], Zhe Wang[3], Guosheng Lin[1,2]*
[1]S-Lab, Nanyang Technological University
[2]School of Computer Science and Engineering, Nanyang Technological University
[3]SenseTime Research
E-mail: `ruibo001@e.ntu.edu.sg` , `gslin@ntu.edu.sg`

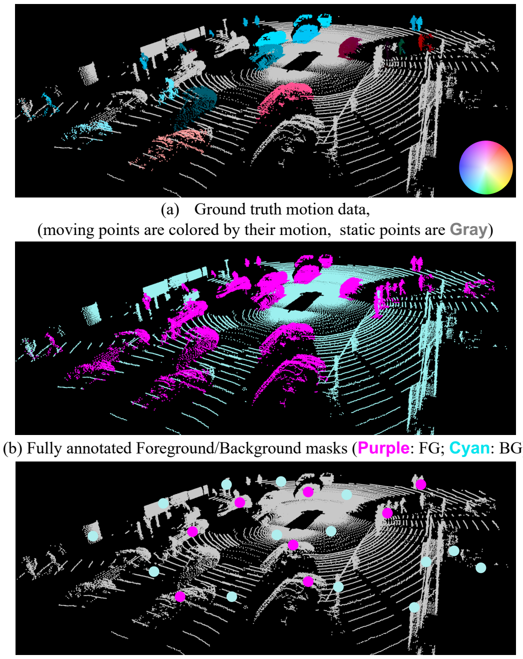
Abstract

Understanding the motion behavior of dynamic environments is vital for autonomous driving, leading to increasing attention in class-agnostic motion prediction in LiDAR point clouds. Outdoor scenes can often be decomposed into mobile foregrounds and static backgrounds, which enables us to associate motion understanding with scene parsing. Based on this observation, we study a novel weakly supervised motion prediction paradigm, where fully or partially (1%, 0.1%) annotated foreground/background binary masks are used for supervision, rather than using expensive motion annotations. To this end, we propose a two-stage weakly supervised approach, where the segmentation model trained with the incomplete binary masks in Stage1 will facilitate the self-supervised learning of the motion prediction network in Stage2 by estimating possible moving foregrounds in advance. Furthermore, for robust self-supervised motion learning, we design a Consistency-aware Chamfer Distance loss by exploiting multi-frame information and explicitly suppressing potential outliers. Comprehensive experiments show that, with fully or partially binary masks as supervision, our weakly supervised models surpass the self-supervised models by a large margin and perform on par with some supervised ones. This further demonstrates that our approach achieves a good compromise between annotation effort and performance.

1. Introduction

Understanding the dynamics of surrounding environments is vital for autonomous driving [27]. Particularly, motion prediction, which generates the future positions of objects from previous information, plays an important role in path planning and navigation.

Classical approaches [7, 9, 47] achieve motion prediction by object detection, tracking, and trajectory forecasting. These detection-based approaches may fail when encountering unknown categories not included in training data [42]. To address this issue, many approaches [8,41,42] propose to directly estimate class-agnostic motion from bird's eye view (BEV) map of point clouds and achieve a good trade-off between accuracy and computational cost. However, sensors can not capture motion information in complex environments [27], which makes motion data scarce and expensive. Therefore, most existing real-world motion data are produced by semi-supervised learning methods with auxiliary information, e.g., KITTI [10, 27], or bootstrapped from human-annotated object detection and



(a) Ground truth motion data,
(moving points are colored by their motion, static points are Gray)

(b) Fully annotated Foreground/Background masks (Purple: FG; Cyan: BG)

(c) Partially annotated Foreground/ Background masks

Figure 1. Illustration of our weak supervision concept. Outdoor scenes can be decomposed into mobile foregrounds and static backgrounds, which enables us to achieve motion learning with fully or partially annotated FG/BG masks as weak supervision to replace expensive ground truth motion data.

[*] Corresponding author: G. Lin. (e-mail: `gslin@ntu.edu.sg`)

tracking data, e.g., Waymo [16]. To circumvent the dependence on motion annotations, PillarMotion [26] utilizes point clouds and camera images for self-supervised motion learning. Although achieving promising results, there is still a large performance gap between the self-supervised method, PillarMotion, and fully supervised methods.

Outdoor scenes can often be decomposed into moving objects and backgrounds [27], which enables us to associate motion understanding with scene parsing. As shown in Fig. 1 (a) and (b), with ego-motion compensation, motion only exists in foreground points. Therefore, if we distinguish the mobile foregrounds from the static backgrounds, we can focus on mining valuable dynamic motion supervision from these potentially moving foreground objects, leading to more effective self-supervised motion learning. Based on this intuition, we propose a novel weakly supervised paradigm, where expensive motion annotations are replaced by fully or partially (1%, 0.1%) annotated foreground/background (FG/BG) masks to achieve a good compromise between annotation effort and performance. To this end, we design a two-stage weakly supervised motion prediction approach, where we train a FG/BG segmentation network with partially annotated masks in Stage1 and train a motion prediction network in Stage2. Specifically, in Stage2, the segmentation network from Stage1 will generate foreground points for training samples, so that the motion prediction network can be trained on these foreground points in a self-supervised manner.

In self-supervised 3D motion learning [18, 26, 44], Chamfer distance (CD) is preferred. However, the CD is sensitive to outliers [37]. Unfortunately, outliers are common in our setting. This is partly due to the view-changes, occlusions, and noise of point clouds and also due to the possible errors in the FG points estimated by the FG/BG segmentation network. To alleviate the impact of outliers, we propose a novel Consistency-aware Chamfer Distance (CCD) loss. Different from the typical CD loss, our CCD loss exploits supervision from multi-frame point clouds and leverages multi-frame consistency to measure the confidence of points. By assigning uncertain points lower weights, our CCD loss suppresses potential outliers.

Our main contributions can be summarized as follows:

- Without using expensive motion data, we propose a weakly supervised motion prediction paradigm with fully or partially annotated foreground/background (FG/BG) masks as supervision to achieve a good compromise between annotation effort and performance. To the best of our knowledge, this is the first work on weakly supervised class-agnostic motion prediction.

- By associating motion understanding with scene parsing, we present a two-stage weakly supervised motion prediction approach, where the FG/BG segmentation generated from Stage1 will facilitate the self-supervised motion learning in Stage2.

- We design a novel Consistency-aware Chamfer Distance loss, where multi-frame information is used to suppresses potential outliers for robust self-supervised motion learning.

- With FG/BG masks as weak supervision, our weakly supervised models outperform the self-supervised models by a large margin, and performs on par with some supervised ones.

2. Related Work

Motion prediction. Classical approaches achieve motion prediction by detecting potential traffic participants and estimating their future trajectories [2,6,7,9,12,24,30,46,47]. However, the object detectors used in these approaches may impair the performance of trajectory prediction, especially, when the detectors encounter unknown categories not contained in training data. To predict class-agnostic motion in open-set traffic scenarios, many recent works [8, 19, 26, 32, 41, 42] attempt to represent the 3D environments with bird's eye view (BEV) map of point clouds. PillarFlow [19] applies a 2D flow structure, PWC-Net [35], to establish correlations on BEV embeddings for motion estimation. MotionNet [42] and BE-STI [41] propose to jointly predict semantic categories and future motion from BEV features. In this work, we adopt the networks designed in MotionNet [42] as our backbone modules.

To overcome the reliance on motion annotations, PillarMotion [26] proposes a self-supervised method, where Chamfer distance loss and 2D optical flow from camera images are used for training. However, there is still a large performance gap to supervised approaches. Different from PillarMotion, our work explores weakly supervised motion prediction, where only foreground/background masks are used as weak supervision to achieve a good compromise between annotation effort and performance. Additionally, we propose a consistency-aware Chamfer loss, which is more robust to outliers than the typical Chamfer loss used in [26].

Scene flow estimation. Scene flow estimation [38], which aims to produce a 3D motion field, could be a reasonable alternative to reason about the class-agnostic motion. However, the huge computational cost of the most scene flow networks [3,5,11,20–22,25,40,44] hinders their applicability in real-time autonomous driving scenario.

Weakly supervised scene flow estimation methods [5,11] are related to our work. Compared with them, our work is different in three aspects: (1) the purpose is different. The goal of our work is to forecast future motion based on past and current observations, but these methods focus on estimating current motion; (2) our supervision is much weaker than theirs. Our work can achieve weakly supervised motion learning with partially (1%, 0.1%) annotated FG/BG masks, however, these methods rely on fully annotated masks; (3) we design a novel Consistency-aware Chamfer loss for motion learning by exploiting multi-frame information and suppressing outliers, which exhibits better robustness than the Chamfer loss used in [5, 11]. In particular, although the most self-supervised and weakly supervised scene flow methods [5, 11, 21, 22, 44] alleviate the dependence on ground truth data, the scene flow networks adopted in those methods may fail to run in real-time.

3. Problem Formulation of Weakly Supervised Motion Prediction

For a sequence of consecutive LiDAR sweeps, following previous works [41, 42], we first synchronize all point clouds to the current frame. Each synchronized point cloud at frame τ is denoted as $\boldsymbol{P}_\tau = \{\boldsymbol{p}_\tau(i) \in \mathbb{R}^3\}_{i=1}^{N_\tau}$, where N_τ is the number of points. Then, we quantize $\boldsymbol{P}_\tau$ into regular 3D voxels $\boldsymbol{V}_\tau \in \{0,1\}^{H\times W\times C}$, where 0 represents empty voxel, 1 represents non-empty voxel, and H, W, C are the numbers of voxels along X, Y, Z axis. By treating the binary vector along the Z axis as features, $\boldsymbol{V}_\tau$ can be viewed as a bird's eye view (BEV) map of size $H \times W$.

Given the current BEV map at frame t and T past BEV maps, $\{\boldsymbol{V}_\tau\}_{\tau=t}^{t-T}$, the task of motion prediction aims to produce a BEV future motion field $\boldsymbol{X}_{\text{mot},t} \in \mathbb{R}^{H\times W\times 2}$ for the frame t, where each element describes the motion of this cell to its corresponding position at next timestamp. Furthermore, by assigning the motion of each cell to all points within this cell, we map $\boldsymbol{X}_{\text{mot},t}$ to point level and get per-point motion prediction $\boldsymbol{F}_t \in \mathbb{R}^{N_t\times 3}$, where the vertical motion is set to zero. This process is formulated as:

$$\boldsymbol{F}_t = \boldsymbol{U}_t[\boldsymbol{X}_{\text{mot},t}; \vec{\mathbf{0}}], \tag{1}$$

where $\boldsymbol{U}_t \in \{0,1\}^{N_t\times HW}$ is the assignment matrix derived from the spatial relationship between $\boldsymbol{P}_t$ and $\boldsymbol{V}_t$.

In our weakly supervised setting, without motion data, we study how to use fully or partially (1%, 0.1%) annotated foreground/background (FG/BG) masks for motion learning. Specifically, in our work, the partially annotated points are randomly sampled from each point cloud.

4. Method

As shown in Fig. 2, our weakly supervised motion prediction approach contains two stages. In Stage1, we train a FG/BG segmentation network, PreSegNet, using partially annotated FG/BG masks as supervision. In Stage2, we train a motion prediction network, WeakMotionNet, with two output heads: a motion prediction head and an auxiliary FG/BG segmentation head. In the training of motion prediction head, for each training sample $\{\boldsymbol{V}_\tau\}_{\tau=t}^{t-T}$, we first select three consecutive point clouds from the past (-1), current (0) and future (+1) timestamps. And then, we use the trained PreSegNet to generate FG/BG points for the three frames. Based on the generated FG/BG points, we employ a novel Consistency-aware Chamfer loss function for self-supervised motion learning. In the training of auxiliary FG/BG segmentation head, we also adopt partially annotated FG/BG masks as supervision. The training of the two heads in WeakMotionNet is performed simultaneously.

In this section, we first revisit the Chamfer loss in 3D motion tasks (Sec. 4.1) and then discuss the details about our proposed Consistency-aware Chamfer Distance loss (Sec. 4.2). Finally, we will introduce the architecture of the two networks and present their training strategies (Sec. 4.3).

4.1. Preliminaries: Chamfer Loss in 3D Motion

Chamfer Distance (CD) is widely used in various point cloud tasks, such as completion [43], generation [45], reconstruction [4], and 3D motion perception [5, 13, 15, 18, 23, 26, 28, 31, 39, 44]. Given two consecutive point sets $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$, and the predicted per-point motion $\boldsymbol{SF}$ from models, the Chamfer Distance loss for self-supervised 3D motion learning can be defined as:

$$\begin{aligned} \widehat{\boldsymbol{S}}_{\boldsymbol{1}} &= \boldsymbol{S_1} + \boldsymbol{SF}, \\ \mathcal{L}_{CD}(\widehat{\boldsymbol{S}}_{\boldsymbol{1}}, \boldsymbol{S_2}) &= \frac{1}{|\widehat{\boldsymbol{S}}_{\boldsymbol{1}}|}\sum_{\boldsymbol{x}\in\widehat{\boldsymbol{S}}_{\boldsymbol{1}}}\min_{\boldsymbol{y}\in\boldsymbol{S_2}}||\boldsymbol{x}-\boldsymbol{y}||_2^2 + \frac{1}{|\boldsymbol{S_2}|}\sum_{\boldsymbol{y}\in\boldsymbol{S_2}}\min_{\boldsymbol{x}\in\widehat{\boldsymbol{S}}_{\boldsymbol{1}}}||\boldsymbol{y}-\boldsymbol{x}||_2^2, \end{aligned} \tag{2}$$

where $\widehat{\boldsymbol{S}}_{\boldsymbol{1}}$ is the warped first point set using the predicted motion. By minimizing the CD between $\widehat{\boldsymbol{S}}_{\boldsymbol{1}}$ and $\boldsymbol{S_2}$, the models learn to predict the motion that moves the first point set toward the second set.

4.2. Consistency-aware Chamfer Distance Loss

In motion prediction, point clouds are synchronized and motion only exists in foreground points. Therefore, in Stage2, we use the trained PreSegNet from Stage1 to generate possible foreground (FG) and background (BG) points of training samples, and train the motion prediction head of WeakMotionNet on the potentially moving foreground points for more effective self-supervised motion learning.

In self-supervised motion learning, Chamfer Distance (CD) loss could be a choice. However, the CD is sensitive to outliers [37]. Unfortunately, outliers are quite common in this task. To alleviate the impact of outliers, we propose a novel **C**onsistency-aware **C**hamfer **D**istance (CCD) loss function. Compared with the original CD loss (Eq. (2)), Our CCD loss is improved in three aspects. (1) Our CCD minimizes not only the distance between the forward warped current data and the future data, but also the distance between the backward warped current data and the past data. Therefore, our CCD can mine supervision from multi-frame information. (2) Our CCD employs multi-frame consistency to measure the confidence of points and assigns uncer-

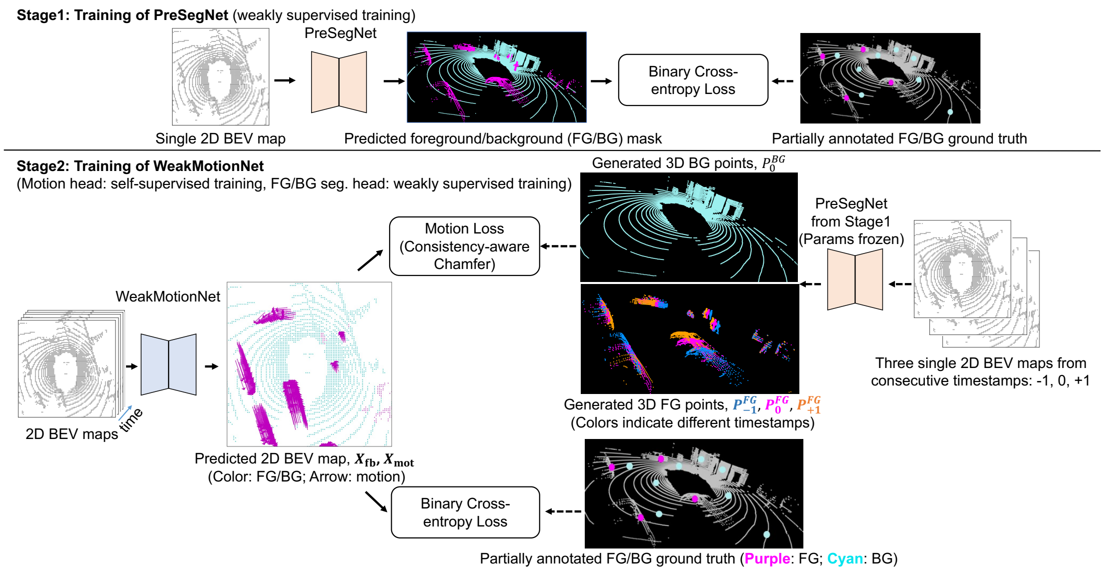

Stage1: Training of PreSegNet (weakly supervised training)
PreSegNet
Single 2D BEV map
Predicted foreground/background (FG/BG) mask
Binary Cross-entropy Loss
Partially annotated FG/BG ground truth
Stage2: Training of WeakMotionNet
(Motion head: self-supervised training, FG/BG seg. head: weakly supervised training)
Generated 3D BG points, P_0^{BG}
Motion Loss (Consistency-aware Chamfer)
PreSegNet from Stage1 (Params frozen)
WeakMotionNet
2D BEV maps
time
Three single 2D BEV maps from consecutive timestamps: -1, 0, +1
Generated 3D FG points, P_{-1}^{FG}, P_0^{FG}, P_{+1}^{FG}
(Colors indicate different timestamps)
Predicted 2D BEV map, X_{fb}, X_{mot}
(Color: FG/BG; Arrow: motion)
Binary Cross-entropy Loss
Partially annotated FG/BG ground truth (Purple: FG; Cyan: BG)

Figure 2. Overview of our two-stage weakly supervised motion prediction approach. In **Stage1**, we train a foreground/background (FG/BG) segmentation network, PreSegNet, with partially annotated masks. In **Stage2**, we train a motion prediction network, WeakMotionNet, which takes a sequence of synchronized BEV maps as input and predicts FG/BG category $\boldsymbol{X}_{\text{fb}}$ and future motion displacement $\boldsymbol{X}_{\text{mot}}$ of each cell. Without motion data, we generate the FG/BG points by the trained PreSegNet from Stage1 and employ a Consistency-aware Chamfer loss with the generated points to train the motion prediction head of WeakMotionNet in a self-supervised manner.

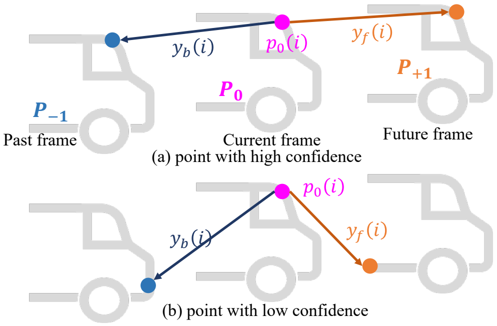


Figure 3. Illustration of confidence estimation. Confidence of each point is measured by the consistency between its forward and backward pseudo motion labels.

tain points fewer weights to suppress potential outliers. (3) Our CCD adopt L_1-norm to calculate the distance between two point clouds, making CCD more robust to outliers.

For each training sample, we denote the generated foreground points from the past (-1), current (0) and future (+1) timestamps as $\boldsymbol{P}_{-1}^{\text{FG}}, \boldsymbol{P}_{0}^{\text{FG}}, \boldsymbol{P}_{+1}^{\text{FG}}$, respectively, and denote the predicted motion of the generated foreground points in the current timestamp as $\boldsymbol{F}_0^{\text{FG}}$. And the per-point motion $\boldsymbol{F}^{\text{FG}}$ is obtained by mapping the predicted BEV motion field $\boldsymbol{X}_{\text{mot}}$ into point level. Note that for simplicity, we omit the FG in this subsection.

Warping the predicted foreground points. Supposing that the motion of objects is consistent within a short temporal window, we obtain the forward warped FG points in current frame $\widehat{\boldsymbol{P}}_{0,f}$ by warping the predicted current FG points $\boldsymbol{P}_0$ with their predicted motion $\boldsymbol{F}_0$, and obtain the backward warped FG points $\widehat{\boldsymbol{P}}_{0,b}$ by warping $\boldsymbol{P}_0$ with the inverse of their predicted motion -$\boldsymbol{F}_0$:

$$\widehat{\boldsymbol{P}}_{0,f} = \boldsymbol{P}_0 + \boldsymbol{F}_0, \quad \widehat{\boldsymbol{P}}_{0,b} = \boldsymbol{P}_0 - \boldsymbol{F}_0. \tag{3}$$

Estimating the confidence of points. The CD loss minimizes the distance between the warped current data and the future data. Specifically, for a point, the CD loss finds its closest point in the other point cloud as correspondence, and uses the coordinate difference as pseudo label to approximate motion ground truth of this point.

A reliable data point should have a consistent pseudo motion label within a short time window. Based on this intuition, given point clouds from three consecutive timestamps, our CCD generates forward and backward pseudo labels and uses the consistency to measure the confidence of this point. An example is shown in Fig. 3. By reweighting our loss function with the confidence, data points with consistent pseudo labels will dominate the training and the outliers will be suppressed. Table 4 shows the effectiveness of the confidence reweighting and Fig. 5 provides a visualization example. The confidence generation for each point in $\boldsymbol{P}_0$ can be formulated as follows:

$$\boldsymbol{y}_f(i) = \arg\min_{\boldsymbol{s}\in\boldsymbol{P}_{+1}} \|\boldsymbol{s} - \widehat{\boldsymbol{p}}_{0,f}(i)\|_2 - \boldsymbol{p}_0(i), \tag{4}$$

$$\boldsymbol{y}_b(i) = \arg\min_{\boldsymbol{s}\in\boldsymbol{P}_{-1}} \|\boldsymbol{s} - \widehat{\boldsymbol{p}}_{0,b}(i)\|_2 - \boldsymbol{p}_0(i), \tag{5}$$

$$w_0(i) = \exp\left(\frac{-\|\boldsymbol{y}_f(i) + \boldsymbol{y}_b(i)\|_2^2}{2\theta^2}\right). \tag{6}$$

In Eq. (4), for a point $\boldsymbol{p}_0(i)$ in $\boldsymbol{P}_0$, we find the closest point in $\boldsymbol{P}_{+1}$ to $\widehat{\boldsymbol{p}}_{0,f}(i)$ as the correspondence of $\boldsymbol{p}_0(i)$, and take the coordinate difference as the forward pseudo label

$\boldsymbol{y}_f(i)$. Based on the same strategy, in Eq. (5), we also obtain its backward pseudo label $\boldsymbol{y}_b(i)$. After that, by taking the consistency between $\boldsymbol{y}_f(i)$ and $\boldsymbol{y}_b(i)$ as a metric, we use a Gaussian kernel to generate its confidence score $w_0(i)$ in Eq. (6). In our experiments, we set θ^2 to 0.5.

According to the confidence map $\boldsymbol{w}_0$ for $\boldsymbol{P}_0$, the confidence map for $\boldsymbol{P}_{+1}$ and $\boldsymbol{P}_{-1}$ can be obtained by nearest search. Here is an example of generating confidence score for a point $\boldsymbol{p}_{+1}(j)$ in $\boldsymbol{P}_{+1}$:

$$I_{+1}(j) = \arg \min_{i \in \{1,\dots,N_0\}} \|\boldsymbol{p}_{+1}(j) - \widehat{\boldsymbol{p}}_{0,f}(i)\|_2, \tag{7}$$

$$w_{+1}(j) = w_0(I_{+1}(j)). \tag{8}$$

In Eq. (7), for a point $\boldsymbol{p}_{+1}(j)$ in $\boldsymbol{P}_{+1}$, we find its closest point in $\widehat{\boldsymbol{P}}_{0,f}$ and get the index of this closest point, $I_{+1}(j)$. Then, in Eq. (8), we take the confidence score of its closest point as the confidence score $w_{+1}(j)$ for $\boldsymbol{p}_{+1}(j)$.

Formulation. The Consistency-aware Chamfer Distance (CCD) loss function can be written as:

$$\begin{aligned} \mathcal{L}_{CCD}(\boldsymbol{P}_{-1}, \boldsymbol{P}_0, \boldsymbol{P}_{+1}, \boldsymbol{F}) = \mathcal{L}_{SCCD}(\widehat{\boldsymbol{P}}_{0,b}, \boldsymbol{P}_{-1}, \boldsymbol{w}_0, \boldsymbol{w}_{-1}) \\ + \mathcal{L}_{SCCD}(\widehat{\boldsymbol{P}}_{0,f}, \boldsymbol{P}_{+1}, \boldsymbol{w}_0, \boldsymbol{w}_{+1}), \end{aligned} \tag{9}$$

where the first term minimizes the distance between the backward warped current points and the past points, and the second term minimizes the distance between the forward warped current points and the future points. Taking the second term as an example, the L_{SCCD} can be formulated as:

$$\begin{aligned} \mathcal{L}_{SCCD}(\widehat{\boldsymbol{P}}_{0,f}, \boldsymbol{P}_{+1}, \boldsymbol{w}_0, \boldsymbol{w}_{+1}) = \frac{1}{\|\boldsymbol{w}_0\|_1} \sum_{i=1}^{N_0} w_0(i) \min_{\boldsymbol{s} \in \boldsymbol{P}_{+1}} \|\widehat{\boldsymbol{p}}_{0,f}(i) - \boldsymbol{s}\|_1 \\ + \frac{1}{\|\boldsymbol{w}_{+1}\|_1} \sum_{j=1}^{N_{+1}} w_{+1}(j) \min_{\boldsymbol{s} \in \widehat{\boldsymbol{P}}_{0,f}} \|\boldsymbol{p}_{+1}(j) - \boldsymbol{s}\|_1. \end{aligned} \tag{10}$$

$\mathcal{L}_{SCCD}$ can be viewed as a weighted Chamfer loss, with confidence map as weight to suppress potential outliers and L_1-norm as metric to measure distance.

4.3. Network Implementation

4.3.1 Pre-segmentation Network (PreSegNet)

PreSegNet is a foreground/background (FG/BG) segmentation model with a backbone network and a FG/BG segmentation head. For the backbone network, we adopt the backbone structure in Motionnet [42] and remove the temporal convolution in each block to make it fit for single frame segmentation. For the FG/BG segmentation head, we adopt two-layer 2D convolutions.

Training. In each frame τ, the point cloud $\boldsymbol{P}_\tau$ is quantized into a single BEV map $\boldsymbol{V}_\tau$, and the PreSegNet takes $\boldsymbol{V}_\tau$ as input and predicts its FG/BG category map $\boldsymbol{X}_{\text{fb},\tau}$. In our weakly supervised setting, the ground truth labels are only available in a tiny fraction of points in $\boldsymbol{P}_\tau$. To train the PreSegNet with incomplete point-wise supervision, we first map $\boldsymbol{X}_{\text{fb},\tau}$ to point level and get per-point category predictions $\boldsymbol{B}_{\text{fb},\tau}$. This process is formulated as: $\boldsymbol{B}_{\text{fb},\tau} = \boldsymbol{U}_\tau \boldsymbol{X}_{\text{fb},\tau}$, where $\boldsymbol{U}_\tau$ is a 0–1 assignment matrix derived from the spatial relationship between $\boldsymbol{P}_\tau$ and $\boldsymbol{V}_\tau$. Then, the classification loss in Stage1 can be written as:

$$\mathcal{L}_{\text{cls}} = \frac{1}{|\mathcal{R}_\tau|} \sum_{i \in \mathcal{R}_\tau} \alpha_\tau(i) \cdot \text{CE}(\boldsymbol{b}_{\text{fb},\tau}(i), \boldsymbol{b}^{\text{gt}}_{\text{fb},\tau}(i)), \tag{11}$$

where $\mathcal{R}_\tau$ is the set of labeled points in $\boldsymbol{P}_\tau$, $\text{CE}(\cdot)$ is a cross-entropy loss, $\boldsymbol{b}_{\text{fb},\tau}(i)$ is the predicted FG/BG category of point i, and $\boldsymbol{b}^{\text{gt}}_{\text{fb},\tau}(i)$ is its ground truth label. Specifically, $\alpha_\tau(i)$ is the weight assigned to different categories. $\alpha_\tau(i)$ is 0.005 if the ground truth label of point i is background (BG); otherwise, 1.

4.3.2 Motion Prediction Network (WeakMotionNet)

WeakMotionNet is a motion prediction network containing a backbone network, a motion prediction head, and an auxiliary FG/BG segmentation head. We implement the backbone network using the same structure as the one in MotionNet [42] and implement the two output heads with two-layer 2D convolutions. More details about PreSegNet and WeakMotionNet are in supplementary.

Training. In each frame t, the WeakMotionNet takes a sequence of synchronized BEV maps $\{\boldsymbol{V}_\tau\}_{\tau=t}^{t-T}$ as input and predicts the future motion map $\boldsymbol{X}_{\text{mot},t}$ and FG/BG category map $\boldsymbol{X}_{\text{fb},t}$ of frame t. Using the assignment matrix $\boldsymbol{U}_t$, we get the point-wise motion $\boldsymbol{F}_t$ and category $\boldsymbol{B}_{\text{fb},t}$, as presented in Eq. (1).

In the training of motion prediction head, we select three consecutive point clouds from the past (t-1), current (t) and future (t+1) timestamps, and use the trained PreSegNet from Stage1 to generate their FG/BG points. Specially, in our experiments, we set the time span between two timestamps to 0.5s. Since the point clouds are synchronized, we treat the generated BG points as static and only apply the CCD loss on the generated FG points, $\boldsymbol{P}^{\text{FG}}_{t\text{-}1}, \boldsymbol{P}^{\text{FG}}_{t}, \boldsymbol{P}^{\text{FG}}_{t+1}$. Therefore, the motion loss contains two parts:

$$\mathcal{L}_{\text{mot}} = \mathcal{L}_{CCD}(\boldsymbol{P}^{\text{FG}}_{t\text{-}1}, \boldsymbol{P}^{\text{FG}}_{t}, \boldsymbol{P}^{\text{FG}}_{t+1}, \boldsymbol{F}^{\text{FG}}_{t}) + \mathcal{L}_{\text{mot,BG}}(\boldsymbol{F}^{\text{BG}}_{t}), \tag{12}$$

where the first term is the CCD loss (Eq. (9)) for the predicted motion of the generated FG points, $\boldsymbol{F}^{\text{FG}}_{t}$, and the second term is for the predicted motion of the generated BG points, $\boldsymbol{F}^{\text{BG}}_{t}$. Regarding the generated BG points as static, we train their predicted motion, $\boldsymbol{F}^{\text{BG}}_{t}$, to be zero:

$$\mathcal{L}_{\text{mot,BG}}(\boldsymbol{F}^{\text{BG}}_{t}) = \frac{1}{N^{\text{BG}}_{t}} \sum_{i=1}^{N^{\text{BG}}_{t}} \|\boldsymbol{f}^{\text{BG}}(i) - \vec{\mathbf{0}}\|_1, \tag{13}$$

where N^{BG}_{t} is the number of the generated BG points.

In the training of auxiliary FG/BG segmentation head, we use the same classification loss (Eq. 11) and follow the same strategy used in Sec. 4.3.1. The total loss for Stage2 is the combination of the two loss functions:

$$\mathcal{L}_{\text{Stage2}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{mot}}. \tag{14}$$

Table 1. Evaluation results of motion prediction on nuScenes test set. Full., Self., Weak., refer to fully-supervised, self-supervised, and weakly supervised training respectively. With fully (100%) or partially (1%, 0.1%) annotated FG/BG masks as weak supervision, our models outperforms the self-supervised model by a large margin, and performs on par with some supervised ones, which demonstrates that our approach achieves a good compromise between annotation effort and performance.

Method	Supervision	Modality	Static		Speed $\leq$ 5m/s		Speed $>$ 5m/s	
			Mean ↓	Median ↓	Mean ↓	Median ↓	Mean ↓	Median ↓
FlowNet3D [25]	Full.	LiDAR	0.0410	0	0.8183	0.1782	8.5261	8.0230
HPLFlowNet [14]	Full.	LiDAR	0.0041	0.0002	0.4458	0.0960	4.3206	2.4881
PointRCNN [34]	Full.	LiDAR	0.0204	0	0.5514	0.1627	3.9888	1.6252
LSTM-ED [33]	Full.	LiDAR	0.0358	0	0.3551	0.1044	1.5885	1.0003
PillarMotion [26]	Full.	LiDAR+Image	0.0245	0	0.2286	0.0930	0.7784	0.4685
MotionNet [42]	Full.	LiDAR	0.0201	0	0.2292	0.0952	0.9454	0.6180
BE-STI [41]	Full.	LiDAR	0.0220	0	0.2115	0.0929	0.7511	0.5413
PillarMotion [26]	Self.	LiDAR+Image	0.1620	0.0010	0.6972	0.1758	3.5504	2.0844
Ours (0.1%)	Weak. (0.1% FG/BG masks)	LiDAR	0.0426	0	0.4009	0.1195	2.1342	1.2061
Ours (1%)	Weak. (1% FG/BG masks)	LiDAR	0.0558	0	0.4337	0.1305	1.7823	1.0887
Ours (100%)	Weak. (100% FG/BG masks)	LiDAR	**0.0243**	**0**	**0.3316**	**0.1201**	**1.6422**	**1.0319**

Table 2. Results of FG/BG segmentation on nuScenes test set.

Method	FG Acc. ↑	BG Acc. ↑	Overall Acc. ↑
Ours (0.1%)	83.5%	**96.0%**	95.2%
Ours (1.0%)	91.0%	95.7%	**95.4%**
Ours (100%)	**93.8%**	94.5%	94.4%

Table 3. Motion prediction results on Waymo Dataset

Method	Supervision	Static	Speed $\leq$ 5m/s	Speed $>$ 5m/s
MotionNet [42]	Full.	0.0263	0.2620	0.9493
Ours (0.1%)	Weak.(0.1% FG/BG masks)	0.0297	0.3581	1.6362
Ours (1.0%)	Weak.(1.0% FG/BG masks)	0.0334	0.3458	**1.5655**
Ours (100%)	Weak.(100% FG/BG masks)	**0.0219**	**0.3385**	1.6576

In inference, we will regularize the final motion predictions by setting the motion of predicted background areas to zero.

5. Experiments

In this section, we first compare our models with SOTA supervised and self-supervised motion prediction methods in Sec. 5.1. And then, we conduct ablation studies to analyze the effectiveness of each component in Sec. 5.2.

Dataset. The main experiments are conducted on nuScenes [1], a large-scale autonomous driving dataset. Following previous works [26, 41, 42], we adopt 500 scenes for training, 100 for validation, and 250 for testing. For each scene, we utilize the LiDAR point clouds as input. In the training stage, We use the officially annotated foreground and background labels as weak supervision. In the validation and testing stage, we generate motion data from detection and tracking annotations provided by nuScenes as ground truth for evaluation. Also, we apply our approach to Waymo Open Dataset [36]. Specifically, we extract 14,351 samples from training set for training and 3,634 from validation set for testing. More details are in supplementary.

Implementation details. Following the same data preprocessing settings in [41, 42], we crop each input point cloud in the range of $[-32, 32] \times [-32, 32] \times [-3, 2]$ meters and set the voxel size to be $(0.25, 0.25, 0.4)\text{m}$ for nuScenes. For Waymo, we set range to $[-32, 32] \times [-32, 32] \times [-1, 4]$.

In Stage1, we train the FG/BG segmentation network, PreSegNet, with partially annotated FG/BG masks as supervision for 40 epochs. We set the batchsize to 16 and use Adam [17] with an initial learning rate of 0.0005, which is decayed by 0.5 after every 10 epochs.

In Stage2, we train the motion prediction network, WeakMotionNet, with a sequence of point clouds as input. For fair comparisons with [26, 41, 42], we set the sequence length to 5. Each input sequence contains 1 current frame and 4 past frames, and the time span between each two consecutive frames is 0.2s. Following [26], the WeakMotionNet is designed to output the displacement for the next 0.5s as the predicted motion. Correspondingly, in self-supervised motion learning, the past frame and the future frame are point clouds in the past 0.5s and the next 0.5s, respectively. And the trained PreSegNet from Stage1 will generate FG and BG points for the past, current, and future frames. Note that, when using fully annotated masks as supervision, we omit the Stage1 and directly use the ground truth FG and BG points for self-supervised motion learning. We train the WeakMotionNet for 60 epochs with an initial learning rate of 0.0005, and we decay it by 0.5 after every 10 epochs. We set the batchsize to 8 and use Adam as optimizer. Our method is implemented in PyTorch [29]. More experimental details are contained in supplementary.

Evaluation metrics. For the motion prediction, following previous works [26, 41, 42], we divide non-empty cells into three groups: static, slow ($\leq$ 5m/s), fast ($\geq$ 5m/s) and evaluate the mean and median errors on each group. Errors are measured by L_2 distances between the predicted displacements and the ground truth displacements for the next 1s. The outputs of our WeakMotionNet are the displacements for the next 0.5s. Therefore, we assume that the speed is constant within a short time windows and linearly interpolate the outputs to the next 1s for evaluation. For the FG/BG segmentation, we measure the accuracy of each category (Acc.) and overall classification accuracy (Overall Acc.), i.e., the average accuracy over all non-empty cells.

5.1. Comparison with State-of-the-Art Methods

In Table 1, we compare our weakly supervised approach with various SOTA motion prediction methods on nuScenes [1]. PillarMotion [26] is the best self-supervised

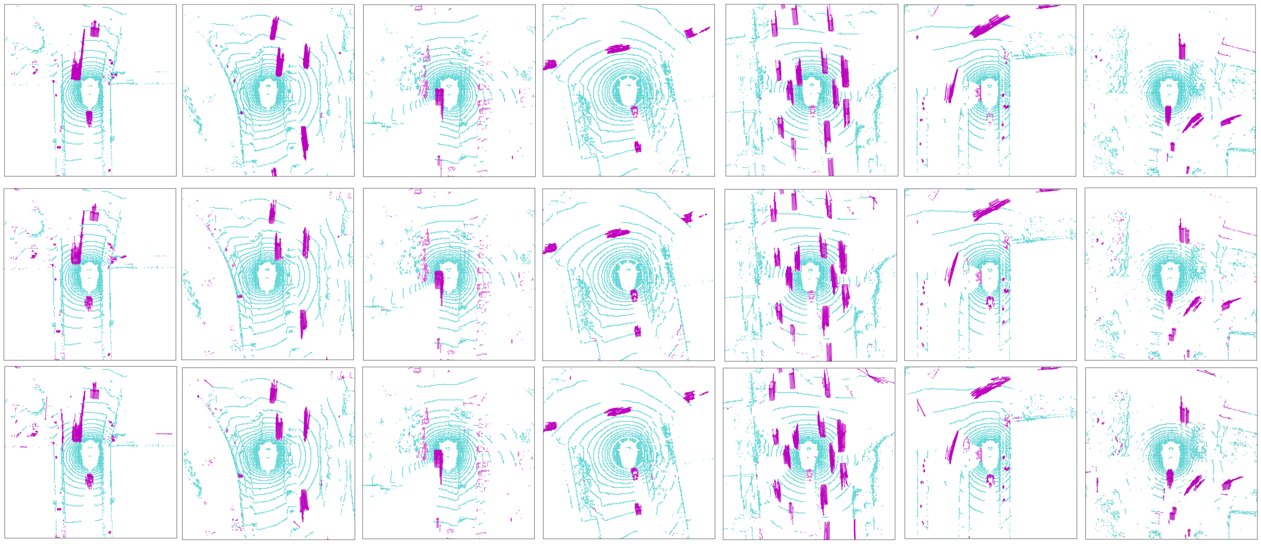


Figure 4. Qualitative results of motion prediction and foreground/background segmentation on nuScenes. Top: ground-truth. Middle: results of our method trained by 100% annotated FG/BG masks. Bottom: results of our method trained by 1% annotated masks. We show motion with an arrow attached to each cell and represent different category with different color. **Purple**: Foreground; **Cyan**: Background.

Table 4. Ablation study for Consistency-aware Chamfer Distance (CCD) loss under the FG/BG annotation ratio of 1%.

Loss function in WeakMotionNet	L2-norm	L1-norm	Future Frame	Past Frame	Confidence Reweight	Auxiliary FG/BG Segmentation	Static	Speed ≤ 5m/s	Speed > 5m/s
								Mean Error ↓	
Chamfer loss (**Baseline**)	✓		✓				0.4416	0.8087	2.3981
Chamfer-L1		✓	✓				0.2579 (−42%)	0.5110 (−37%)	2.1229 (−11%)
Multi-frame Chamfer-L1		✓	✓	✓			0.2677 (−39%)	0.5240 (−35%)	**1.7436 (−27%)**
Consistency-aware Chamfer		✓	✓	✓	✓		<u>0.1469</u> (−67%)	<u>0.4390</u> (−46%)	<u>1.7729</u> (−26%)
Consistency-aware Chamfer + Seg. (**Ours, 1%**)		✓	✓	✓	✓	✓	**0.0558 (−87%)**	**0.4337 (−46%)**	1.7823 (−26%)

method, which utilizes an off-the-shelf optical flow estimation network and additional 2D images for training. Without using any knowledge from images or optical flow, our models trained by 1% or 0.1% annotated FG/BG masks outperform the self-supervised PillarMotion by about **35%** on all evaluation metrics. Comparing our weakly supervised models with fully supervised models, we observe that our models perform better than FlowNet3D [25], HPLFlowNet [14], and PointRCNN [34] on both slow and fast speed groups. Especially, our models outperform fully supervised scene flow models, FlowNet3D and HPLFlowNet, by about **70%** and **50%** on the fast speed group, respectively. The comparisons show that our weakly supervised approach achieves a good compromise between annotation effort and performance and reduces the gap to fully supervised approaches.

The performance of the FG/BG segmentation head of WeakMotionNet is shown in Table 2. Despite being trained with a tiny fraction of annotated masks (1% or 0.1%), our models can distinguish foreground and background with high overall accuracy (about 94%). Qualitative results are shown in Fig. 4. More visualization results are in supplementary. With a sequence of BEV maps as input, our WeakMotionNet takes 16ms for inference in a RTX A5000 GPU.

For further evaluation, we also apply our weakly supervised approach to Waymo [36]. As presented in Table 3, the mean errors of our weakly supervised models with different annotation ratios are less than 0.04m, 0.4m and 1.7m on static, slow, and fast groups, respectively. FG/BG segmentation and visualization results are in supplementary.

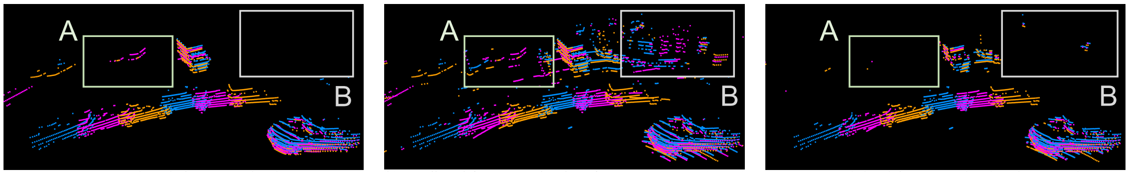
Table 5. Impact of different data format in CCD loss

Data format in CCD loss	Static	Speed ≤ 5m/s	Speed > 5m/s
2D BEV	0.0587	0.5302	2.8176
3D Point (Ours, 1%)	**0.0558**	**0.4337**	**1.7823**

5.2. Ablation Studies

In this subsection, we evaluate the effectiveness of our approach on nuScenes.

Ablation study for Consistency-aware Chamfer Distance loss. For robust self-supervised motion learning, we design a Consistency-aware Chamfer loss with L_1-norm as distance metric, multi-frame point clouds for supervision, and multi-frame consistency for reweighting. As presented in Table 4, compared with the Chamfer loss, the baseline method, our Chamfer-L1 loss with L_1-norm as distance metric reduces the prediction error on the three groups by 42%, 37%, and 11%, respectively. When adding point clouds from the past frame as part of the target data, our multi-frame Chamfer-L1 loss further decreases the error on the fast speed group from 2.12m to 1.74m. Moreover, on the basis of the multi-frame loss, by using multi-frame consistency for reweighting, our Consistency-aware Chamfer loss drops the error by an additional 28% and 11% for the static and slow groups, respectively. The results in Table 4



(a) Ground truth foreground points (b) Predicted foreground points from PreSegNet (0.1%) (c) Reweighted foreground points by CCD loss

Figure 5. Visualization for PreSegNet and CCD loss. Outliers may be due to occlusions of points (e.g., region A), and inaccurate foreground predictions from PreSegNet (e.g., region B). In our CCD loss, we use multi-frame consistency to measure the confidence of points and assign uncertain points fewer weights, thereby suppressing potential outliers. For better visualization, we remove points with lower weights in (c). Different color represents point cloud in different frames. **Blue**: past frame; **Purple**: current frame; **Orange**: future frame.

Table 6. Effectiveness of two-stage training framework

Method	Static	Speed ≤ 5m/s	Speed > 5m/s
1% masks w/o Stage1	1.1976	3.1904	8.9025
1% masks with Stage1 (Ours)	**0.0558**	**0.4337**	**1.7823**

indicate that our Consistency-aware Chamfer loss achieves substantial improvements compared to the Chamfer loss.

Furthermore, to regularize the predicted motion, we also use an auxiliary FG/BG segmentation head for WeakMotionNet and set the motion of predicted background areas to zero. As shown in Table 4, by combining our Consistency-aware Chamfer loss with a FG/BG segmentation loss, we observe a significantly lower error on static group. In supplementary, we further provide an ablation of the CCD loss and the segmentation loss under different annotation ratios.

In the training of WeakMotionNet, we map the predicted BEV motion field $\boldsymbol{X}_{\mathrm{mot},t}$ to point level, and apply our loss to the point-wise motion predictions $\boldsymbol{F}_t$ and 3D point clouds for self-supervised motion learning. In Table 5, we compare our design with an alternative approach, where we directly apply the loss function to BEV motion field and BEV maps. As presented in Table 5, mapping predictions to point level and applying the loss to 3D points works best, which demonstrates the effectiveness of our design.

Ablation study for two-stage training framework. To enable weakly supervised learning with partially annotated FG/BG masks, we design a two-stage framework, where a FG/BG segmentation network, PreSegNet, in Stage1 is trained with these incomplete masks and further generates dense FG/BG masks to facilitate the self-supervised motion learning of WeakMotionNet in Stage2. The ablation results for two-stage framework are in Table 6. Without using PreSegNet from Stage1, an alternative approach is to generate dense FG/BG masks from the segmentation head of WeakMotionNet online during training. As shown in Table 6, this alternative approach performs significantly worse than our framework. This may be because, during training, the undertrained segmentation head is more likely to generate inaccurate FG/BG masks, thereby hindering self-supervised motion learning. Table 7 presents the results of our PreSegNet on nuScenes validation set. Despite being trained with 1% or 0.1% annotated masks, our PreSegNet still achieves a good segmentation accuracy of foreground areas (about 93%). Qualitative results of PreSegNet are in supplementary. Accurate foreground segmentation from PreSegNet makes our CCD loss able to be applied in most mobile objects, ensuring the performance of moving objects.

Table 7. Results of foreground/background segmentation produced by PreSegNet in Stage1 on nuScenes validation set.

Method	FG Acc. ↑	BG Acc. ↑	Overall Acc. ↑
PreSegNet (0.1% FG/BG masks)	93.1%	89.5%	89.7%
PreSegNet (1.0% FG/BG masks)	**94.6%**	**92.0%**	**92.2%**

Visualization for PreSegNet and CCD loss. In Fig. 5, we provide a visualization example of training data. In our weakly supervised motion prediction, outliers may be due to occlusion of points (e.g., region A), and inaccurate foreground predictions from PreSegNet (e.g., region B), which may further impair the training. To address this issue, in our CCD loss, we use multi-frame consistency to measure the confidence of points and assign uncertain points fewer weights, thereby suppressing potential outliers. As shown in Fig. 5(c), the number of outliers in region A and B is reduced. More visualization examples are in supplementary.

6. Conclusion

In this work, we study weakly supervised motion prediction with FG/BG masks as supervision. Specifically, we present a two-stage approach and a Consistency-aware Chamfer Distance (CCD) loss. Experiments show that our weakly supervised models surpass self-supervised ones and perform on par with some supervised ones, yielding a good compromise between annotation effort and performance.

Limitations. (1) The self-supervised motion learning in Stage2 relies on the FG/BG segmentation from Stage1, which makes inaccurate segmentation hinder motion learning. (2) The CCD loss may fail to handle large displacements. More discussions are in supplementary.

Acknowledgments. This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). This research is partly supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-003), and the MoE AcRF Tier 2 grant (MOE-T2EP20220-0007) and MoE AcRF Tier 1 grants (RG14/22, RG95/20).

References

[1] 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. 6

[2] 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. 2

[3] Wencan Cheng and Jong Hwan Ko. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *European Conference on Computer Vision*, pages 108–124. Springer, 2022. 2

[4] Jaesung Choe, Byeongin Joung, Francois Rameau, Jaesik Park, and In So Kweon. Deep point cloud reconstruction. *arXiv preprint arXiv:2111.11704*, 2021. 3

[5] Guanting Dong, Yueyi Zhang, Hanlin Li, Xiaoyan Sun, and Zhiwei Xiong. Exploiting rigidity constraints for lidar scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12776–12785, 2022. 2, 3

[6] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 2

[7] 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

[8] 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. 1, 2

[9] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 1, 2

[10] 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

[11] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly supervised learning of rigid 3d scene flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5692–5703, 2021. 2, 3

[12] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021. 2

[13] Xiaodong Gu, Chengzhou Tang, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Rcp: Recurrent closest point for point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8216–8226, 2022. 3

[14] 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. 6, 7

[15] Pan He, Patrick Emami, Sanjay Ranka, and Anand Rangarajan. Learning scene dynamics from point cloud sequences. *International Journal of Computer Vision*, 130(3):669–695, 2022. 3

[16] 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*, 7(2):1589–1596, 2021. 2

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4114–4123, 2021. 2, 3

[19] 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. 2

[20] Ruibo Li, Guosheng Lin, Tong He, Fayao Liu, and Chunhua Shen. Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 364–373, 2021. 2

[21] Ruibo Li, Guosheng Lin, and Lihua Xie. Self-point-flow: Self-supervised scene flow estimation from point clouds with optimal transport and random walk. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15577–15586, 2021. 2, 3

[22] Ruibo Li, Chi Zhang, Guosheng Lin, Zhe Wang, and Chunhua Shen. Rigidflow: Self-supervised scene flow learning on point clouds by local rigidity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16959–16968, 2022. 2, 3

[23] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 34:7838–7851, 2021. 3

[24] 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 *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11553–11562, 2020. 2

[25] 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. 2, 6, 7

[26] 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. 2, 3, 6

[27] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 1, 2

[28] Bojun Ouyang and Dan Raviv. Occlusion guided self-supervised scene flow estimation on 3d point clouds. In *2021 International Conference on 3D Vision (3DV)*, pages 782–791. IEEE, 2021. 3

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[30] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020. 2

[31] Jhony Kaesemodel Pontes, James Hays, and Simon Lucey. Scene flow from point clouds with or without learning. In *2020 international conference on 3D vision (3DV)*, pages 261–270. IEEE, 2020. 3

[32] 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. 2

[33] 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. 6

[34] 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. 6, 7

[35] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 2

[36] 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. 6, 7

[37] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3405–3414, 2019. 2, 3

[38] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729. IEEE, 1999. 2

[39] Chaoyang Wang, Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural prior for trajectory estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6532–6542, 2022. 3

[40] Guangming Wang, Yunzhe Hu, Zhe Liu, Yiyang Zhou, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. What matters for 3d scene flow network. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 38–55. Springer, 2022. 2

[41] Yunlong Wang, Hongyu Pan, Jun Zhu, Yu-Huan Wu, Xin Zhan, Kun Jiang, and Diange Yang. Be-sti: Spatial-temporal integrated network for class-agnostic motion prediction with bidirectional enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17093–17102, 2022. 1, 2, 3, 6

[42] 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, 5, 6

[43] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*, 2021. 3

[44] 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. 2, 3

[45] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 3

[46] 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. 2

[47] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904. PMLR, 2021. 1, 2