Self-Supervised Bird's Eye View Motion Prediction with Cross-Modality Signals

Shaoheng Fang[1], Zuhong Liu[1], Mingyu Wang[2], Chenxin Xu[1], Yiqi Zhong[3], Siheng Chen[1,4]
[1] Shanghai Jiao Tong University
[2] University of Chinese Academy of Sciences
[3] University of Southern California
[4] Shanghai AI Laboratory
{shfang, xcxwakaka, sihengc}@sjtu.edu.cn, zuhong.liu@polytechnique.edu,
wangmingyu21a@mails.ucas.ac.cn, yiqizhon@usc.edu

Abstract

Learning the dense bird's eye view (BEV) motion flow in a self-supervised manner is an emerging research for robotics and autonomous driving. Current self-supervised methods mainly rely on point correspondences between point clouds, which may introduce the problems of fake flow and inconsistency, hindering the model's ability to learn accurate and realistic motion. In this paper, we introduce a novel cross-modality self-supervised training framework that effectively addresses these issues by leveraging multi-modality data to obtain supervision signals. We design three innovative supervision signals to preserve the inherent properties of scene motion, including the masked Chamfer distance loss, the piecewise rigidity loss, and the temporal consistency loss. Through extensive experiments, we demonstrate that our proposed self-supervised framework outperforms all previous self-supervision methods for the motion prediction task.

Introduction

Accurate prediction of dynamic motion within a scene is fundamental for the safe and robust planning of autonomous vehicles. Instead of predicting instance-level trajectories (Chen et al. 2020). An emerging trend is to predict the dense motion flow in the BEV (Bird's Eye View) map directly from raw sequential sensor input in an end-to-end manner (Wu, Chen, and Metaxas 2020; Wang et al. 2022). This approach is less susceptible to perception errors and possesses the capability to discern class-agnostic motion (Wu, Chen, and Metaxas 2020; Wong et al. 2020). Nevertheless, training flow prediction models with supervision necessitates a substantial volume of annotations for sensor data and annotating motion labels for sensor data proves to be intricate and costly. Hence, the effective utilization of vast amounts of unlabeled raw data for motion prediction training has emerged as a notable and encouraging challenge. Recently, many works have proposed various self-supervised frameworks to learn the BEV motion without relying on ground truth labels (Luo, Yang, and Yuille 2021; Li et al. 2023; Jia et al. 2023).

Inspired by self-supervised scene flow estimation, current self-supervised BEV motion prediction methods (Luo, Yang, and Yuille 2021; Li et al. 2023) primarily rely on chamfer distance loss to establish the point-level correspondences between point clouds. However, this heavy dependence on point-level correspondences leads to two major problems when learning motion patterns from real-world LiDAR point cloud data.

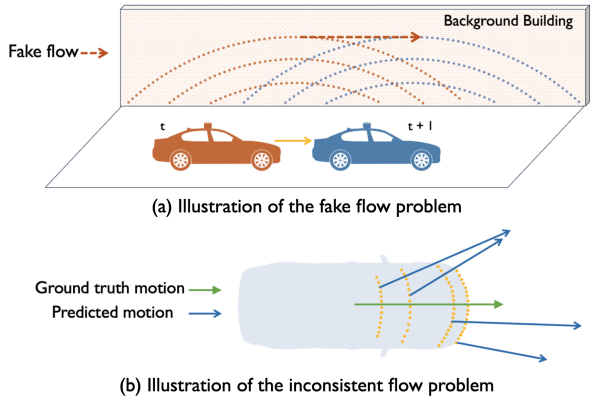


Figure 1: Problems in current self-supervised motion learning methods that rely on point correspondence. (a) For static objects (background building), points with correspondences in the point cloud sequence may have completely different locations, misleading the model to learn the fake flow. (b) Due to the sparse nature of the point cloud, points within an instance may learn highly varying flow.

The first problem is fake flows. Due to the alterations in the viewpoint of the LiDAR sensor, points associated with the background or static objects often exhibit flow that does not exist, as shown in Figure 1(a). This fake flow will mislead the model to learn incorrect motion patterns, thereby adversely affecting the accuracy of predictions. One previous work (Li et al. 2023) introduces a weakly supervised setting where foreground/background ground truth is available to mitigate the impact of noise originating from background points to alleviate the problem. However, the method still remains limited as extra human annotation is indispensable.

The second problem is the inconsistent flows within one single object; see an illustration in Figure 1(b). Owing to the inherent sparsity in point cloud data, the point-level flows associated with the same objects may exhibit inconsistent motions when solely relying on the point correspondences. This problem of inconsistency violates the object-level rigid con-

straints and causes confusion for model learning procedures. (Luo, Yang, and Yuille 2021) aims to preserve the local uniformity of motion flow by employing a smoothness loss, which encourages minimal changes among neighbor flow values. Unfortunately, this assumption fails in the boundary region between moving objects and the background and is unable to ensure instance-level motion consistency.

To address the challenges of fake flow and inconsistent flow, rather than developing another network model, our focus is to design dedicated supervision signals to preserve a series of inherent properties of scene motion. For the fake flow issue, one common, yet fundamental property is that motion is restricted to moving objects. Stationary components should exhibit no motion, allowing us to filter out background noise and obtain a more precise motion flow. To tackle the inconsistent flow challenge, we focus on two primary properties: object and temporal consistency. In essence, points within rigid objects should move uniformly. Furthermore, object motions should remain relatively stable over short periods, ensuring no abrupt changes. By emphasizing these key properties, we enhance the uniformity and reliability of the motion flow.

However, due to the notorious noise and sparsity issues of point cloud data, relying solely on point cloud sequences might compromise the accurate representation of scene motion properties. To compensate, we leverage the multi-modality information. We specifically incorporate sequential camera images—readily accessible since most robots are equipped with cameras, thereby incurring no extra annotation costs. These sequential images enable the extraction of optical flow, providing a rich layer of motion insights. This stands in stark contrast to the sparse, irregular, and fragmented data in point cloud sequences. Optical flow images distinctly highlight the coherent and consistent motion of objects, sharply separating them from the background.

Leveraging the advantages of multi-modality data, we incorporate the spirit of preserving scene motion's inherent properties into a novel self-supervised training framework for BEV motion prediction. Specifically, i) to ensure that motion is exclusive to moving objects, the framework generates a pseudo static/dynamic mask for each point cloud according to the optical flow data. Then this mask will be used to ensure structural consistency exclusively for the dynamic portion through a novel masked Chamfer distance loss; ii) to promote motion consistency in individual objects, we employ a simple clustering technique to the optical flow image, discerning instance boundaries and creating pixel clusters in the image space. Then the cluster information for each pixel will be projected to the point cloud space, creating rigid point cloud clusters that should share the same motion flow to ensure the instance-level rigidity constraints; and iii) for temporal motion consistency, we introduce a novel temporal consistency loss, which enforces the smoothness of predictions across long point cloud sequences. Note that image data are only used for providing supervision signals in the training phase; during inference, the proposed BEV motion prediction network only needs point cloud sequences.

Experimental evaluations conducted on the nuScenes (Caesar et al. 2020) dataset demonstrate that our proposed methodology improves upon previous self-supervised approaches by up to 40%. Notably, our method achieves performance comparable to weakly-supervised and fully-supervised methods.

To summarize, the main contributions of our work are:

- We propose a novel cross-modality self-supervised training framework for BEV motion prediction, which leverages multi-modality data to obtain supervision signals.
- We propose three novel supervision signals to preserve the inherent properties of scene motion, including the masked Chamfer distance loss, the piecewise rigidity loss and the temporal consistency loss.
- Our method achieves state-of-the-art performance. Comprehensive experiments demonstrate the effectiveness of our designed framework.

Related Work

Motion Prediction

The goal of motion prediction is to estimate the future movements of mobile objects in a scene based on past observations. Traditional approaches tackle this issue via a two-stage framework, relying on the results of 3D object detection and tracking to predict the instance-level trajectories (Casas et al. 2020; Luo, Yang, and Urtasun 2018; Phillips et al. 2021). However, the dependence on intermediate results may lead to error accumulation and a limited ability to perceive unknown classes (Wu, Chen, and Metaxas 2020; Wong et al. 2020). An emerging trend is to predict dense future motion in an end-to-end framework directly from sequential sensor input, including multi-frame point clouds (Wu, Chen, and Metaxas 2020; Lee et al. 2020; Luo, Yang, and Yuille 2021; Filatov, Rykov, and Murashkin 2020; Wang et al. 2022; Wei et al. 2023) and multi-view images (Hu et al. 2021; Zhang et al. 2022; Fang et al. 2023).

Training a motion prediction model requires high-quality manual labels, but obtaining such labels is both expensive and laborious. Accordingly, some methods aim to mitigate this issue from various perspectives. (Luo, Yang, and Yuille 2021) proposes a self-supervision method that utilizes point cloud structure consistency and cross-modality regularization; (Li et al. 2023) proposes the use of a weakly supervised setting that only utilizes foreground/background information, effectively improving the accuracy. (Jia et al. 2023) employs contrastive learning to learn BEV pillar features and uses pillar association to predict motion. In this paper, we propose a novel self-supervised framework and achieve remarkable performance that is comparable to other weakly supervised and even fully supervised approaches.

Self-Supervised Scene Flow Estimation

Scene flow estimation aims to determine the 3D motion displacement at the point level between a pair of point clouds (Liu, Qi, and Guibas 2019; Puy, Boulch, and Marlet 2020; Jund et al. 2021; Cheng and Ko 2022; Li et al. 2021; Gu et al. 2019). Learning scene flow in a self-supervised manner is a popular field of research (Mittal, Okorn, and Held 2020; Wu et al. 2019; Baur et al. 2021; Kittenplon, Eldar, and Raviv 2021; Tishchenko et al. 2020). (Mittal,

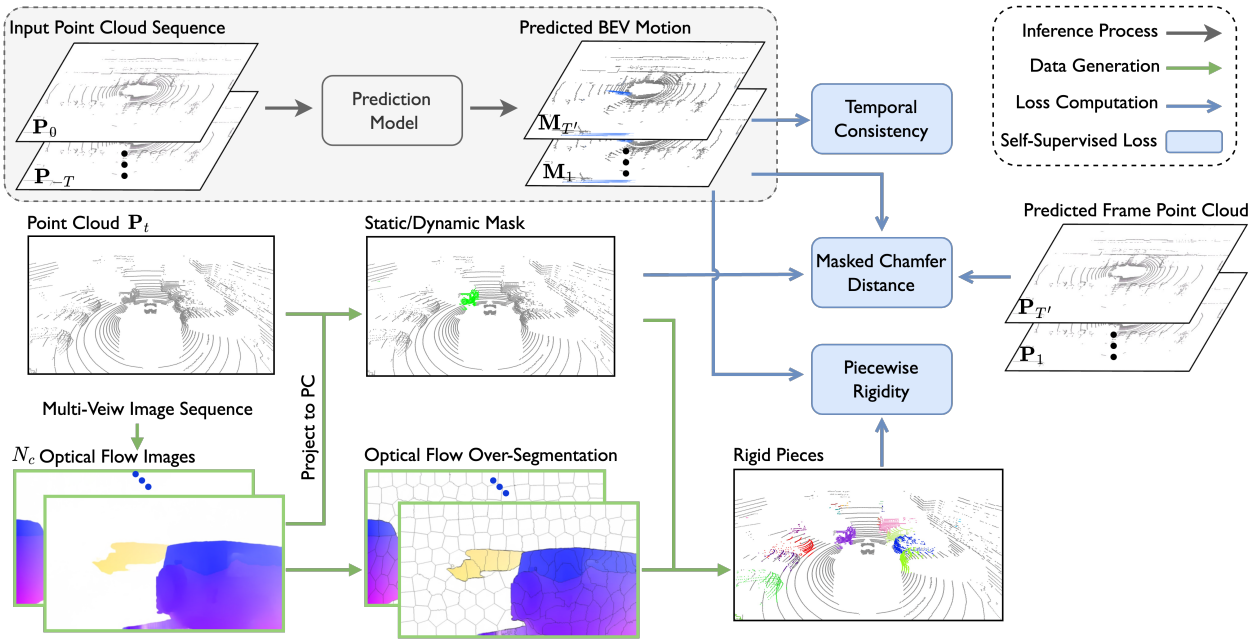


Figure 2: An overview of our cross-modality self-supervision learning framework. An overview of our cross-modality self-supervision learning framework. For self-supervised training, we introduce three innovative self-supervised losses that align with real-world motion patterns. The inference process only takes the point cloud sequence as input and predicts the motion flow of each BEV cell (grey area).

Okorn, and Held 2020) was the first to establish a self-supervised learning framework that utilizes a combination of nearest neighbor and cycle consistency loss. Following (Wu et al. 2019), (Kittenplon, Eldar, and Raviv 2021; Pontes, Hays, and Lucey 2020) use the chamfer distance loss to learn the point correspondences between two point clouds. (Li et al. 2022b; Gojcic et al. 2021) employ ego-motion estimation and exploit the piecewise rigid nature of point clouds.

We follow the philosophy of (Wu et al. 2019; Li et al. 2022b) by designing self-supervised loss to maintain structural consistency between point clouds and exploiting the piecewise rigidity for regularization. Nevertheless, most of these methods (Li et al. 2022b; Wu et al. 2019; Mittal, Okorn, and Held 2020) usually assume strong one-to-one correspondences between point clouds and incur heavy computational costs, making them unsuitable for real-time perception in autonomous driving. We propose a masked chamfer loss to mitigate these issues. Moreover, unlike scene flow estimation, which identifies motion between a pair of point clouds, we concentrate on predicting the future of the scene based on point cloud sequences.

LiDAR-Camera Fusion

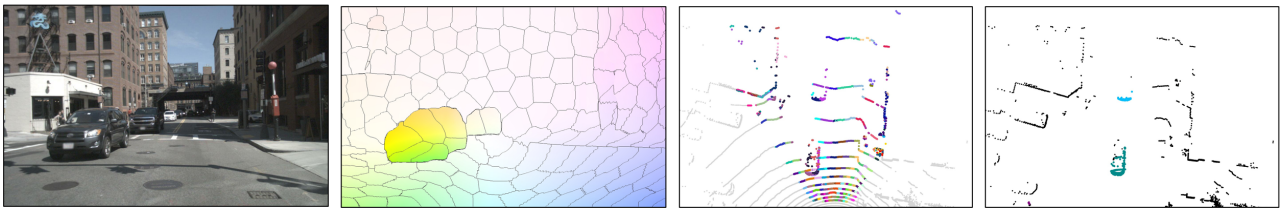
LiDAR-camera fusion has been extensively investigated to enhance scene perception, including various tasks such as 3D object detection (Vora et al. 2020; Li et al. 2022c; Liang et al. 2022) and scene flow estimation (Rishav et al. 2020; Liu et al. 2022). A novel line of research is to leverage cross-modality information as supervised signals to support model training. (Ding et al. 2023) combines detection and tracking results from LiDAR point clouds with odometry data and optical flow to jointly improve radar scene flow learning. (Li et al. 2022a) generates noisy pseudo-labels from optical flow to supervise scene flow learning. Additionally, (Luo, Yang, and Yuille 2021) facilitates motion prediction learning through LiDAR-camera cross-modality regularization. Optical flow data, which can be easily obtained from camera video without human labeling, has shown the potential to aid motion learning on point clouds. However, these methods solely employ the numerical values of optical flow as the guidance for point cloud motion and ignore the inherent advantages of optical flow data over point clouds.

Method

This section introduces a self-supervised training framework for BEV motion prediction, where three novel supervision signals are generated from multi-modality inputs, including point cloud sequences and camera videos.

Problem Formulation

The objective of the motion prediction task is to directly forecast the motion of mobile grids in the 3D BEV map from historical point cloud sequences (Li et al. 2023; Wang et al. 2022). The prediction model takes the current frame 0 along with T past frames of point clouds that synchronized to the current frame as input. The point cloud sequence is denoted as $\mathbf{P}_t = \{p_i^t \in \mathbb{R}^3\}_{i=1}^{N_t},\ t = 0, -1, \cdots, -T$, where N_t represents the number of points in P_t. The multi-view camera video is utilized in the training process. The corresponding multi-view images of $\mathbf{P}_t$ are $\{\mathbf{I}_t^k \in \mathbb{R}^{H\times W\times 3}\}_{k=1}^{N_c}$, where N_c is the number of cameras.



(a) (b) (c) (d)

Figure 3: Rigid piece generation. (a) A frame of sequential images; (b) Over-segmentation on the optical flow image; (c) Over-segmentation projected to the associated point cloud; (d) Rigid pieces after fusion. In (c) and (d), each color refers to a piece.

The future motion is represented in the form of a BEV (bird's eye view) map. Assuming the model predicts T' frames of future motion field, $\mathbf{M}_t \in \mathbb{R}^{X\times Y\times 2}$, $t = 1, \cdots, T'$ represents the motion field of the t frame, where $X \times Y$ is the shape of the BEV map according to the vehicle-ego coordinates at the current timestamp. Considering that each grid in the BEV map represents a rather small area in the real-world scene, points within the same pixel grid have identical motion flow. To generate a point-level 3D motion flow, each point can be assigned the motion of its corresponding position in the BEV map based on its 3D coordinates, and the vertical motion is set as zero. The motion of points is denoted as $\mathbf{F}_t = \left\{f_i^t \in \mathbb{R}^3\right\}_{i=1}^{N_0}$, $t = 1, \cdots, T'$.

Apart from predicting the future motion from time 1 to T', the model can also infer the motion situation from the current frame to the past frames. Therefore, in the following section, $\mathbf{T} = \{t_1, \cdots, t_n\}$ is used to represent the entire set of time frames that the model predicts, including 1 to T' and possible backward predictions (for example, -1).

Overview

Figure 2 overviews our training framework. Since the prediction model is not our focus, we directly adopt MotionNet (Wu, Chen, and Metaxas 2020). The key of this work is to leverage multi-modality inputs to provide three supervision signals that can preserve inherent properties of scene motion. They include: 1) pseudo static/dynamic mask loss generated from sequential video, 2) the piece-wise rigidity loss, and 3) temporal motion consistency loss. Here are the detailed descriptions.

Pseudo Static/Dynamic Mask Loss

We employ Chamfer Distance as the foundation of learning structural consistency. The Chamfer Distance serves as a measure of similarity between two sets of points. In self-supervised training (Wu et al. 2019; Kittenplon, Eldar, and Raviv 2021) or optimization (Li, Kaesemodel Pontes, and Lucey 2021; Pontes, Hays, and Lucey 2020) methods related to point clouds, the chamfer distance loss is a widely used technique that helps maintain the structural consistency of two point clouds.

For any frame t, with the predicted point-level flow $\mathbf{F}_t$ from frame 0 to frame t, the predicted point cloud can be calculated as $\mathbf{P}'_t = \{p'_i \in \mathbb{R}^3 \mid p'_i = p_i^0 + f_i^t\}_{i=1}^{N_0}$. Given the point cloud P_t at frame t, the self-supervised chamfer distance loss can be defined as

$$\mathcal{L}_{cd}(\mathbf{P}_t, \mathbf{P}'_t) = \sum_{p_j\in\mathbf{P}_t} \min_{p'_i\in\mathbf{P}'_t} \left\|p_j - p'_i\right\|_2^2 + \sum_{p'_i\in\mathbf{P}'_t} \min_{p_j\in\mathbf{P}_t} \left\|p'_i - p_j\right\|_2^2. \tag{1}$$

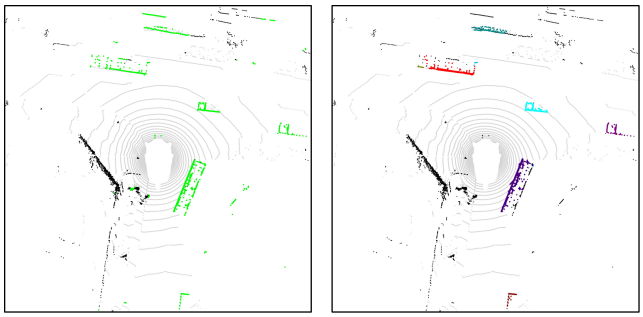
However, the point cloud data is often sparse and full of noise points. Even for stationary objects, the point cloud representation can vary significantly with the sensor's movement (Khurana et al. 2023). This poses great challenges and introduces noise when relying on Chamfer distance loss for learning. To better understand the motion of a dynamic 3D scene, it is crucial to focus on moving targets while disregarding the background and stationary objects. However, due to the sparse and noisy nature of the point cloud, it is difficult to distinguish between the static and dynamic parts of a point cloud in open scenes. In contrast, optical flow in the image space is much more accessible and easier to obtain. Video data is abundant with superior temporal and texture information, and the relevant techniques are already well-established (Sun et al. 2018; Teed and Deng 2020). Previous works (Luo, Yang, and Yuille 2021; Ding et al. 2023) have utilized the value of image optical flow to assist in learning point cloud scene flow. In our method, we propose to extract a pseudo static/dynamic mask from the optical flow results of the image data to aid in structure consistency learning.

Given the point cloud time frame $t \in \mathbf{T}$, we can get the adjacent image pairs $(\mathbf{I}_t^k, \mathbf{I}_{t+\delta t}^k)$, $k = 1, \cdots, N_c$ from the camera video. For brevity, we omit the superscript k for camera index and the subscript t for frame index in subsequent contents, and use $\mathbf{I}$ and $\mathbf{I}'$ to denote $\mathbf{I}_t^k$ and $\mathbf{I}_{t+\delta t}^k$. The optical flow generated from $\mathbf{I}$ and $\mathbf{I}'$ is denoted as $\mathbf{F}^{\text{2D}} \in \mathbb{R}^{H\times W\times 2}$.

The optical flow $\mathbf{F}^{\text{2D}}$ cannot yet be directly used to determine the motion status of each pixel. Apart from the optical flow generated by dynamic targets in the scene, the movement of the ego vehicle also produces flow in the camera view. Following (Luo, Yang, and Yuille 2021), we divide the optical flow into two parts, $\mathbf{F}^{\text{2D}} = \mathbf{F}_{ego}^{\text{2D}} + \mathbf{F}_{mot}^{\text{2D}}$, where $\mathbf{F}_{ego}^{\text{2D}}$ corresponds to the optical flow caused by vehicle motion and $\mathbf{F}_{mot}^{\text{2D}}$ corresponds to the optical flow caused by dynamic objects.

The numerical value of $\mathbf{F}_{ego}^{\text{2D}}$ can be calculated through the sensors' poses. Let $p_i \in \mathbf{P}$ represent a point within the image $\mathbf{I}$ and $\mathcal{T}_{\mathbf{P}\to\mathbf{I}}$ represent the transformation matrix from the lidar point cloud $\mathbf{P}$ to the image $\mathbf{I}$

$$(u_i, v_i) = \mathcal{T}_{\mathbf{P}\to\mathbf{I}}(p_i). \tag{2}$$



(a) (b)

Figure 4: An example of the generated static/dynamic mask and the rigid piece labels. Left: green represents dynamic points while black represents static points; Right: each color except black refers to a rigid piece label.

The value of $\mathbf{F}^{\text{2D}}_{ego}$ corresponding to p_i is then

$$\mathbf{F}^{\text{2D}}_{ego}(u_i, v_i) = \mathcal{T}_{\mathbf{P}\to\mathbf{I}'}(p_i) - \mathcal{T}_{\mathbf{P}\to\mathbf{I}}(p_i). \tag{3}$$

Ideally, the calculated $\mathbf{F}^{\text{2D}}_{mot}$ corresponding to a stationary target or background would be close to 0. Thus, static points can be distinguished by setting a small threshold for $\mathbf{F}^{\text{2D}}_{mot}$.

Nevertheless, when dealing with distant moving objects that are far from the camera, their corresponding optical flow values may be small and incorrectly classified as static. To mitigate such effect, we employ the projected 3D scene flow to supplement the static assessment. Denote $\mathbf{F}^{\text{2D}}_{mot}(u_i, v_i)$ as f_i^{2D}. With the constraint of zero vertical motion, we can project the 2D optical flow f_i^{2D} to a 3D scene flow originating from p_i. The operation is represented by a projection $\mathcal{T}_{\text{optf}\to\text{sf}}$ (see more info in supp.).

$$f_i^{\text{3D}} = \mathcal{T}_{\text{optf}\to\text{sf}}(f_i^{\text{2D}}). \tag{4}$$

The pseudo static/dynamic status s_i of p_i is estimated as

$$s_i = \begin{cases} 0, & f_i^{\text{2D}} < \tau^{\text{2D}} \text{ and } f_i^{\text{3D}} < \tau^{\text{3D}}, \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

A pseudo static/dynamic mask $\mathbf{S}_t \in \mathbb{R}^{N_t}$ is produced for the point cloud $\mathbf{P}_t$ at each time frame. Utilizing $\mathbf{S}_t$, $\mathbf{P}_t$ can be separated into two parts: a pseudo dynamic point cloud $\tilde{\mathbf{P}}_t$ and a pseudo static point cloud $\bar{\mathbf{P}}_t$. The Chamfer distance loss calculation is then performed on the pseudo dynamic point cloud instead of the entire point cloud. The masked Chamfer loss can be defined as

$$\mathcal{L}_{mc} = \frac{1}{|\mathbf{T}|}\sum_{t\in\mathbf{T}}\left(\mathcal{L}_{cd}(\tilde{\mathbf{P}}_t, \tilde{\mathbf{P}}'_t) + \mathcal{L}_{static}(\bar{\mathbf{P}}_0, t)\right), \tag{6}$$

where $\mathcal{L}_{cd}(\cdot)$ is the Chamfer loss and

$$\mathcal{L}_{static}(\bar{\mathbf{P}}_0, t) = \frac{1}{|\bar{\mathbf{P}}_0|}\sum_{p_i\in\bar{\mathbf{P}}_0} ||f_i^t||_1, \tag{7}$$

which pushes the motions of static points to be zero.

Piecewise Rigidity Loss

When considering flow estimation on point clouds, local rigidity is an important physical prior that is frequently utilized (Dong et al. 2022; Gojcic et al. 2021; Li et al. 2022b; Shen et al. 2023). Unlike previous self-supervised methods for point clouds that maintain rigidity by clustering on a single-frame point cloud (Li et al. 2022b; Shen et al. 2023), we introduce a simple yet effective method for point cloud piecewise rigidity based on optical flow image clustering. Compared to a single-frame point cloud, optical flow images exhibit several desirable properties. The motion consistency of dynamic objects is more pronounced in optical flow images and is easier to recognize and extract. In contrast to the sparsity and noise of point clouds, the pixels of moving objects in optical flow images are adjacent, with high uniformity and smoothness. Additionally, the boundaries between objects and the background are more salient.

For the optical flow image $\mathbf{F}^{\text{2D}}$, we apply a simple image clustering method (Achanta et al. 2012) to obtain over-segmentation labels (Figure 3(b)). The over-segmentation divides the entire image into numerous small pieces. Due to the optical flow consistency of moving objects, it is easy to ensure that almost all pixels in a single small piece belong to the same object.

Through the point cloud to image projection function $\mathcal{T}_{\mathbf{P}\to\mathbf{I}}$, we can retrieve the corresponding label in the image over-segmentation for each point p_i. By projecting the over-segment result labels from N_c camera views, we obtain an over-segmentation result on the whole point cloud.

Some points in the point cloud may be occluded from the camera view due to the slight difference between the LiDAR sensor position and the camera sensor position. For each segment piece, we consider the distance between all points in the piece and the camera and find the smallest distance d_{min}. We set a threshold Δd such that if a point in the piece is more than $d_{min} + \Delta d$ away from the camera, it will be excluded from the piece.

As a result, we obtain hundreds of rigid pieces from the point cloud (Figure 3(c)). In order to achieve more accurate and unified segmentation, we employ a simple height-based rigid piece fusion method. Since the visual perspective of the multi-view cameras is typically parallel to the ground, rigid pieces derived from images can easily appear at different heights in the same location. Given the assumption that points within the same grid of the BEV map have the same motion, we consolidate multiple rigid piece labels into a singular label if a grid contains points with distinct labels.

Finally, we generate N_r rigid pieces for the point cloud $\mathbf{P_0}$ (Figure 3(d)). $\mathcal{R}_1, \cdots, \mathcal{R}_{N_r}$ are the N_r rigid pieces, where $\mathcal{R}$ represents a set of points that have the same rigid piece label. For any frame $t \in \mathbf{T}$, the piecewise rigidity loss function is defined as

$$\mathcal{L}^t_{pr} = \frac{1}{N_r}\sum_{j=1}^{N_r}\frac{1}{|\mathcal{R}_j|}\sum_{p_i\in\mathcal{R}_j}|f^{mean}_{\mathcal{R}_j} - f_i^t|, \tag{8}$$

where $f^{mean}_{\mathcal{R}_j} = \sum_{p_i\in\mathcal{R}_j} f_i^t/|\mathcal{R}_j|, \quad j = 1, \cdots, N_r$. The final piecewise rigidity loss $\mathcal{L}_{pr}$ is the mean of $\mathcal{L}^t_{pr}$ over all time frames t.

Temporal Motion Consistency Loss

For moving objects in traffic scenes, such as cars, pedestrians, and bicycles, their motion patterns do not undergo significant changes over short periods of time. In a point cloud sequence, the displacement of points belonging to a moving object should remain consistent over equal time intervals. Therefore, for self-supervised learning of point cloud sequences, we can apply point-level temporal consistency constraints to the predicted motion. The temporal consistency loss is defined as

$$\mathcal{L}_{tc} = \frac{1}{N_0} \sum_{i=1}^{N_0} \frac{1}{|\mathbf{T}|} \sum_{t \in \mathcal{T}} |f_i^{mean} - f_i^t|, \tag{9}$$

where $f_i^{mean} = \sum_{t \in \mathcal{T}} f_i^t / (t|\mathbf{T}|), \;\; i = 1, \cdots, N_0$.

Overall Loss

In summary, the total loss for the model training is a weighted sum of the proposed masked Chamfer distance loss, piecewise rigidity loss and temporal consistency regularization.

$$\mathcal{L} = \lambda_{mc} \cdot \mathcal{L}_{mc} + \lambda_{pr} \cdot \mathcal{L}_{pr} + \lambda_{tc} \cdot \mathcal{L}_{tc}, \tag{10}$$

where λ_{mc}, λ_{pr}, and λ_{tc} are the balancing parameters.

Experiments

Experimental Setup

Dataset. We evaluate our approach on the NuScenes (Caesar et al. 2020) dataset. NuScenes contains 1000 scenes, each of which has 20 seconds of LiDAR point cloud sequences and multi-view camera videos annotated at 2Hz. Following the setting in previous works for fair comparisons (Wu, Chen, and Metaxas 2020; Wang et al. 2022; Luo, Yang, and Yuille 2021; Li et al. 2023; Jia et al. 2023), we adopt 500 scenes for training, 100 scenes for validation, and 250 scenes for testing. During training, we utilize both the LiDAR point clouds and camera images, while only LiDAR point cloud data is required for the validation and testing of the model. The ground truth BEV motion flow for validation and testing is generated from the detection and tracking annotation from the NuScenes dataset.

Implementation details. Initially, the BEV feature maps are extracted from the multi-frame point clouds by (Lang et al. 2019). Our model backbone is built upon MotionNet (Wu, Chen, and Metaxas 2020), which takes sequential BEV features as input and extracts spatial-temporal features. The input point clouds are cropped within a range of $[-32, 32] \times [-32, 32] \times [-3, 2]$ meters, and the BEV output map is 256×256 in size, which means each cell has a range of $0.25\text{m} \times 0.25\text{m}$. It is worth noting that our proposed cross-modality self-supervision framework is independent of the network backbone. Also, during the inference process, only sequential point cloud data is needed as the model input.

To generate the optical flow, we employ (Teed and Deng 2020) as the optical flow estimation model with the pre-trained parameters offered by Pytorch. The static/dynamic classification thresholds in eq.5 are $\tau^{2D} = 5$pixels and $\tau^{3D} = 1$m. Besides, we extract the points of the ground plane based on the heights and designate them as the static part of the scene. For the training loss in eq.10, we set $\lambda_{mc} = 1$, $\lambda_{pr} = 0.1$ and $\lambda_{tc} = 0.4$. We employ AdamW (Loshchilov and Hutter 2017) optimization algorithm for training. All models are trained on four NVIDIA 3090 GPUs with a batch size of 64. We train the model for 100 epochs with an initial learning rate of 0.008, and we decay the learning rate by 0.5 every 20 epochs.

Metrics. Following previous works (Wu, Chen, and Metaxas 2020; Wang et al. 2022; Luo, Yang, and Yuille 2021; Li et al. 2023; Jia et al. 2023), we use the mean and median errors of motion flow on non-empty cells for evaluation. The error is computed by the L2 distance between the predicted motion flow and ground truth flow for the next 1s future. The final results are presented in three categories divided by varying speeds: static (background and static objects), slow (speed $\leq$ 5 m/s), and fast (speed $>$ 5m/s). Regarding the whole model, we directly utilize the 1s future flow output to calculate the metrics. In ablation studies, if the model only predicts the subsequent 0.5s of future flow, we employ linear interpolation to estimate the predicted flow for the next 1s future.

Comparison with SOTA Methods

Table 1 presents a comprehensive comparison between our proposed self-supervised approach and other methods for BEV motion prediction. Based on the training supervision, all approaches can be categorized into three groups: fully supervised, weakly supervised, and self-supervised. We see that our method achieves state-of-the-art performance in the self-supervised group and surpasses previous methods by a significant margin. Compared to the previous state-of-the-art method (Jia et al. 2023), we exhibit a remarkable improvement of 41% in fast speed metrics, which represent the more challenging and crucial part of motion prediction, 7% in slow speed metrics, and 38% in static metrics.

(Li et al. 2023) is a weakly supervised method that adopts foreground/background annotation as extra supervision signals. Notably, our method shows comparable performance and even surpasses it in terms of the mean error of fast motion. Furthermore, our method outperforms some fully supervised methods such as (Gu et al. 2019) and (Shi, Wang, and Li 2019) by 52% and 48% respectively.

Ablation Studies

Masked Chamfer loss. To enhance the robustness of self-supervised learning by mitigating the noises in point cloud sequence data, we design a masked Chamfer distance loss based on the pseudo static/dynamic mask generated from optical flow images. An example of the generated static/-dynamic mask is illustrated in Figure 4(a). Exp. 1&3, 2&5 in Table 2 compare the results of the original Chamfer distance loss (eq. 1) with the masked Chamfer distance loss (eq. 6). We can see that the masked Chamfer distance loss can improve all metrics by a large margin. Especially for the static motion metrics, the masked Chamfer distance loss can bring up to 75% improvement. This shows its effectiveness of eliminating noise and disturbances originating from

Method	Supervision	Static		Speed $\leq$ 5 m/s		Speed $>$ 5 m/s	
		Mean↓	Median↓	Mean↓	Median↓	Mean↓	Median↓
HPLFlowNet (Gu et al. 2019)	supervised	0.0041	0.0002	0.4458	0.0960	4.3206	2.4881
PointRCNN (Shi, Wang, and Li 2019)	supervised	0.0204	0	0.5514	0.1627	3.9888	1.6252
LSTM-ED (2019)	supervised	0.0358	0	0.3551	0.1044	1.5885	1.0003
MotionNet (Wu, Chen, and Metaxas 2020)	supervised	0.0201	0	0.2292	0.0952	0.9454	0.6180
PillarMotion (Luo, Yang, and Yuille 2021)	supervised	0.0245	0	0.2286	0.0930	0.7784	0.4685
BE-STI (Wang et al. 2022)	supervised	0.0220	0	0.2115	0.0929	0.7511	0.5413
WeakMotionNet (Li et al. 2023)	weakly sup.	0.0426	0	0.4009	0.1195	2.1342	1.2061
FlowNet3D (Liu, Qi, and Guibas 2019)	pre.	2.0514	0	2.2058	0.3172	9.1923	8.4923
HPLFlowNet (Gu et al. 2019)	pre.	2.2165	1.4925	1.5477	1.1269	5.9841	4.8553
PillarMotion (Luo, Yang, and Yuille 2021)	self.	0.1620	0.0010	0.6972	0.1758	3.5504	2.0844
ContrastMotion (Jia et al. 2023)	self.	0.0829	0	0.4522	**0.0959**	3.5266	1.3233
Ours	self.	**0.0514**	**0**	**0.4212**	0.1073	**2.0766**	**1.3226**

Table 1: Evaluation results of BEV motion prediction on nuScenes (Caesar et al. 2020) test set. There are four kinds of training supervision: supervised, weakly-supervised (weakly sup.), pre-trained (pre.), and self-supervised (self.). Our method outperforms other self-supervised methods by a significant margin.

the static background, which constitutes the majority of the point cloud data.

Piecewise rigidity. To ensure uniformity of motion within the same instance, we design an algorithm to generate instance pieces initially from over-segmentation on optical flow images and propose a piecewise rigidity loss to regulate the motion consistency in each piece. Figure 4(b) provides an illustration of the generated pieces.

Exp. 1&2, 3&5 in Table 2 demonstrate the effectiveness of the piecewise rigidity loss, resulting in an improvement of approximately 15% across all evaluation metrics. Exp.4 in Table 2 utilizes a simple neighborhood smoothness loss to constrain the local rigidity of prediction motion, which serves a similar purpose to our piecewise rigidity approach (see more info in supp.). Exp. 4&5 indicates that our method outperforms the alternative smoothness loss in performance. Moreover, the piecewise rigidity loss brings significant advantages in terms of training time and computational resources.

Temporal consistency. Table 3 presents the results of the ablation study conducted on temporal consistency and prediction frames. It is evident that all experiments incorporating the temporal consistency loss exhibit higher performance, which highlights the effectiveness of temporal consistency as a motion pattern that aids in the learning of motion prediction. Furthermore, we explore the impact of different prediction frames on training a motion prediction model. The complete framework predicts the motion of frames -1, 1, and 2 during training with a time interval of 0.5 seconds, and the temporal consistency loss is applied across all predicted frames. In Table 3, the 'past' frame refers to the backward frame -1 and the 'future' frame refers to frame 2. We see that i) As the number of frames involved in motion prediction learning increases, the prediction performance improves correspondingly. This is because the temporal consistency pattern becomes more prominent over a longer point cloud sequence. ii) Predicting backward motion (frame -1) yields a larger improvement compared to predicting a further future frame (frame 2). Due to the ego vehicle's movement, the variations in point cloud data become larger when the time interval expands, which makes learning the correspondence between point clouds a more challenging task

Exp.	m.c.	smooth.	p.r.	Static	Speed $\leq$ 5 m/s	Speed $>$ 5 m/s
1				0.2515	0.8771	3.4098
2			✓	0.2097	0.7135	3.1892
3	✓			0.0704	0.4815	2.5389
4	✓	✓		0.0677	0.4493	2.2142
5	✓		✓	0.0514	0.4212	2.0766

Table 2: Ablation of masked Chamfer distance and piecewise rigidity losses. m.c.: masked Chamfer distance loss; smooth.: smoothness regularization; p.r.: piecewise rigidity.

past	future	temp.	Static	Speed $\leq$ 5 m/s	Speed $>$ 5 m/s
✓	✓		0.1150	0.5549	2.7503
	✓	✓	0.0748	0.5307	2.8830
✓		✓	0.0838	0.5074	2.1814
✓	✓	✓	0.0514	0.4212	2.0766

Table 3: Ablation for prediction frames and temporal consistency loss. past: backward frame -1; future: frame 2 into the future; temp.: temporal consistency loss.

Please refer to the supplementary materials for more qualitative results and ablation studies.

Conclusions

In this paper, we present a novel cross-modality self-supervised method for BEV motion prediction. Concretely, we exploit static/dynamic classification and rigid pieces on point clouds from sequential multi-view images to facilitate motion learning without any manual annotations. Moreover, we enforce temporal consistency across multiple frames, ensuring temporal smoothness of predicted motion. Comprehensive experiments conducted on the nuScenes dataset demonstrate that our proposed method achieves state-of-the-art performance and all designed modules are effective.

Acknowledgments

This research is supported by NSFC under Grant 62171276 and the Science and Technology Commission of Shanghai Municipal under Grant 21511100900, 22511106101, and 22DZ2229005.

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11): 2274–2282.

Baur, S. A.; Emmerichs, D. J.; Moosmann, F.; Pinggera, P.; Ommer, B.; and Geiger, A. 2021. Slim: Self-supervised lidar scene flow and motion segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13126–13136.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Casas, S.; Gulino, C.; Liao, R.; and Urtasun, R. 2020. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 9491–9497. IEEE.

Chen, S.; Liu, B.; Feng, C.; Vallespi-Gonzalez, C.; and Wellington, C. 2020. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine*, 38(1): 68–86.

Cheng, W.; and Ko, J. H. 2022. Bi-PointFlowNet: Bidirectional Learning for Point Cloud Based Scene Flow Estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, 108–124. Springer.

Ding, F.; Palffy, A.; Gavrila, D. M.; and Lu, C. X. 2023. Hidden gems: 4d radar scene flow learning using cross-modal supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9340–9349.

Dong, G.; Zhang, Y.; Li, H.; Sun, X.; and Xiong, Z. 2022. Exploiting rigidity constraints for lidar scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12776–12785.

Fang, S.; Wang, Z.; Zhong, Y.; Ge, J.; and Chen, S. 2023. TBP-Former: Learning Temporal Bird's-Eye-View Pyramid for Joint Perception and Prediction in Vision-Centric Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1368–1378.

Filatov, A.; Rykov, A.; and Murashkin, V. 2020. 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)*, 9498–9504. IEEE.

Gojcic, Z.; Litany, O.; Wieser, A.; Guibas, L. J.; and Birdal, T. 2021. Weakly supervised learning of rigid 3D scene flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5692–5703.

Gu, X.; Wang, Y.; Wu, C.; Lee, Y. J.; and Wang, P. 2019. 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*, 3254–3263.

Hu, A.; Murez, Z.; Mohan, N.; Dudas, S.; Hawke, J.; Badrinarayanan, V.; Cipolla, R.; and Kendall, A. 2021. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15273–15282.

Jia, X.; Zhou, H.; Zhu, X.; Guo, Y.; Zhang, J.; and Ma, Y. 2023. ContrastMotion: Self-supervised Scene Motion Learning for Large-Scale LiDAR Point Clouds. *arXiv preprint arXiv:2304.12589*.

Jund, P.; Sweeney, C.; Abdo, N.; Chen, Z.; and Shlens, J. 2021. Scalable scene flow from point clouds in the real world. *IEEE Robotics and Automation Letters*, 7(2): 1589–1596.

Khurana, T.; Hu, P.; Held, D.; and Ramanan, D. 2023. Point Cloud Forecasting as a Proxy for 4D Occupancy Forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1116–1124.

Kittenplon, Y.; Eldar, Y. C.; and Raviv, D. 2021. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4114–4123.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.

Lee, K.-H.; Kliemann, M.; Gaidon, A.; Li, J.; Fang, C.; Pillai, S.; and Burgard, W. 2020. Pillarflow: End-to-end birds-eye-view flow estimation for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007–2013. IEEE.

Li, B.; Zheng, C.; Li, G.; and Ghanem, B. 2022a. Learning scene flow in 3d point clouds with noisy pseudo labels. *arXiv preprint arXiv:2203.12655*.

Li, R.; Lin, G.; He, T.; Liu, F.; and Shen, C. 2021. 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*, 364–373.

Li, R.; Shi, H.; Fu, Z.; Wang, Z.; and Lin, G. 2023. Weakly Supervised Class-Agnostic Motion Prediction for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17599–17608.

Li, R.; Zhang, C.; Lin, G.; Wang, Z.; and Shen, C. 2022b. 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, 16959–16968.

Li, X.; Kaesemodel Pontes, J.; and Lucey, S. 2021. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 34: 7838–7851.

Li, Y.; Yu, A. W.; Meng, T.; Caine, B.; Ngiam, J.; Peng, D.; Shen, J.; Lu, Y.; Zhou, D.; Le, Q. V.; et al. 2022c. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17182–17191.

Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; and Tang, Z. 2022. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35: 10421–10434.

Liu, H.; Lu, T.; Xu, Y.; Liu, J.; Li, W.; and Chen, L. 2022. CamLiFlow: Bidirectional camera-LiDAR fusion for joint optical flow and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5791–5801.

Liu, X.; Qi, C. R.; and Guibas, L. J. 2019. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 529–537.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Luo, C.; Yang, X.; and Yuille, A. 2021. Self-supervised pillar motion learning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3183–3192.

Luo, W.; Yang, B.; and Urtasun, R. 2018. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 3569–3577.

Mittal, H.; Okorn, B.; and Held, D. 2020. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11177–11185.

Phillips, J.; Martinez, J.; Bârsan, I. A.; Casas, S.; Sadat, A.; and Urtasun, R. 2021. Deep multi-task learning for joint localization, perception, and prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4679–4689.

Pontes, J. K.; Hays, J.; and Lucey, S. 2020. Scene flow from point clouds with or without learning. In *2020 international conference on 3D vision (3DV)*, 261–270. IEEE.

Puy, G.; Boulch, A.; and Marlet, R. 2020. 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*, 527–544. Springer.

Rishav, R.; Battrawy, R.; Schuster, R.; Wasenmüller, O.; and Stricker, D. 2020. DeepLiDARFlow: A deep learning architecture for scene flow estimation using monocular camera and sparse LiDAR. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10460–10467. IEEE.

Schreiber, M.; Hoermann, S.; and Dietmayer, K. 2019. Long-term occupancy grid prediction using recurrent neural networks. In *2019 International Conference on Robotics and Automation (ICRA)*, 9299–9305. IEEE.

Shen, Y.; Hui, L.; Xie, J.; and Yang, J. 2023. Self-Supervised 3D Scene Flow Estimation Guided by Superpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5271–5280.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 770–779.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8934–8943.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 402–419. Springer.

Tishchenko, I.; Lombardi, S.; Oswald, M. R.; and Pollefeys, M. 2020. Self-supervised learning of non-rigid residual flow and ego-motion. In *2020 international conference on 3D vision (3DV)*, 150–159. IEEE.

Vora, S.; Lang, A. H.; Helou, B.; and Beijbom, O. 2020. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4604–4612.

Wang, Y.; Pan, H.; Zhu, J.; Wu, Y.-H.; Zhan, X.; Jiang, K.; and Yang, D. 2022. 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*, 17093–17102.

Wei, S.; Wei, Y.; Hu, Y.; Lu, Y.; Zhong, Y.; Chen, S.; and Zhang, Y. 2023. Asynchrony-Robust Collaborative Perception via Bird's Eye View Flow. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wong, K.; Wang, S.; Ren, M.; Liang, M.; and Urtasun, R. 2020. Identifying unknown instances for autonomous driving. In *Conference on Robot Learning*, 384–393. PMLR.

Wu, P.; Chen, S.; and Metaxas, D. N. 2020. 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*, 11385–11395.

Wu, W.; Wang, Z.; Li, Z.; Liu, W.; and Fuxin, L. 2019. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*.

Zhang, Y.; Zhu, Z.; Zheng, W.; Huang, J.; Huang, G.; Zhou, J.; and Lu, J. 2022. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*.