Semi-supervised Class-Agnostic Motion Prediction with Pseudo Label Regeneration and BEVMix

Kewei Wang[1,2], Yizheng Wu[1,2], Zhiyu Pan[1], Xingyi Li[1,2], Ke Xian[2], Zhe Wang[3], Zhiguo Cao[1], Guosheng Lin[2*]

[1] Key Laboratory of Image Processing and Intelligent Control, Ministry of Education School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2] S-Lab, Nanyang Technological University
[3] SenseTime Research
{wangkewei, zgcao}@hust.edu.cn, gslin@ntu.edu.sg

Abstract

Class-agnostic motion prediction methods aim to comprehend motion within open-world scenarios, holding significance for autonomous driving systems. However, training a high-performance model in a fully-supervised manner always requires substantial amounts of manually annotated data, which can be both expensive and time-consuming to obtain. To address this challenge, our study explores the potential of semi-supervised learning (SSL) for class-agnostic motion prediction. Our SSL framework adopts a consistency-based self-training paradigm, enabling the model to learn from unlabeled data by generating pseudo labels through test-time inference. To improve the quality of pseudo labels, we propose a novel motion selection and re-generation module. This module effectively selects reliable pseudo labels and re-generates unreliable ones. Furthermore, we propose two data augmentation strategies: temporal sampling and BEVMix. These strategies facilitate consistency regularization in SSL. Experiments conducted on nuScenes demonstrate that our SSL method can surpass the self-supervised approach by a large margin by utilizing only a tiny fraction of labeled data. Furthermore, our method exhibits comparable performance to weakly and some fully supervised methods. These results highlight the ability of our method to strike a favorable balance between annotation costs and performance. Code will be available at https://github.com/kwwcv/SSMP.

Introduction

Understanding the motion behavior within dynamic environments is crucial for a variety of autonomous systems. Traditional methods approach motion perception through trajectory prediction (Chai et al. 2019; Chang et al. 2019; Fang et al. 2020; Liang et al. 2020). However, these approaches may face challenges when handling categories that have not been seen in the training set, mainly due to their reliance on object detection (Wu, Chen, and Metaxas 2020). To address this challenge, class-agnostic motion prediction task (Schreiber, Hoermann, and Dietmayer 2019; Wu, Chen, and Metaxas 2020; Wang et al. 2022; Wei et al. 2022) is proposed to provide complementary information. These methods take a sequence of previous point clouds as input and predict the future displacements for each Bird's Eye View (BEV) cell. Although fully-supervised methods have achieved significant success, they often require substantial amounts of annotated point cloud data, which can be both costly and time-consuming to acquire. To overcome this limitation, self-supervised and weakly supervised approaches have been proposed (Luo, Yang, and Yuille 2021; Li et al. 2023). However, noticeable performance gaps still exist between the best results obtained from these annotation-efficient methods and those achieved by fully-supervised methods. These annotation-efficient methods generate supervision by matching source points with target points, in which handling fast motions becomes challenging (Li et al. 2023). Therefore, we delve into exploiting the use of limited, accurate motion labels to generate supervision for easily accessible unlabeled data, *i.e.*, the semi-supervised motion prediction.

Semi-supervised learning (SSL) seeks to largely alleviate the need for labeled data by leveraging unlabeled data (Berthelot et al. 2019b; Tarvainen and Valpola 2017; Sohn et al. 2020a,b; Zhou et al. 2021; Xu et al. 2021). In this study, we adopt one of the most widely used SSL paradigms, consistency-based self-training (Sohn et al. 2020a). The key idea is to first generate reliable pseudo labels for the unlabeled data (pseudo-labeling), and then train the model to predict the pseudo label when feeding the unlabeled data with perturbation (consistency regularization). Specifically, for the motion prediction task, we specially design two novel data augmentations for consistency regularization and the motion select and re-generate module (MSRM) for pseudo-labeling.

Pseudo-labeling (Lee et al. 2013) plays a role in enhancing semi-supervised learning performance by selecting reliable pseudo-labels while discarding unreliable ones. However, unlike in image classification and object detection, there is currently no appropriate metric like the confidence score to evaluate the reliability of predicted motion. Given that motion represents the displacement from the current cell to the corresponding future cell, an accurate motion would ideally lead the current cell to overlap precisely with the corresponding cell after warping. Based on this fact, we design the motion select and re-generate module (MSRM) to select reliable predicted motions by measuring the distance be-

*Corresponding author.

tween the warped cell and its correspondence. Additionally, labels of the discarded cells are re-generated by the neighbor reliable labels based on the assumption of local smoothness.

Two simple yet efficient data augmentations for motion prediction are proposed as Temporal-Sampling (TS) and BEVMix. These augmentations will serve as strong augmentations to drive the weak-to-strong consistency regularization. Specifically, TS is used to sub-sample the input sequence temporally to generate additional artifact samples with larger motion labels. Meanwhile, BEVMix is designed to mix two different BEV sequences to synthesize new training samples with a large diversity. This diversity can enhance the generalization ability of the model, and therefore improve the prediction performance.

Following previous works (Wu, Chen, and Metaxas 2020; Luo, Yang, and Yuille 2021), we test the efficacy of our method on a large-scale autonomous driving dataset, nuScenes (Caesar et al. 2019). We use 1%,5%, and 10% of labeled data as labeled sets and the remainder as unlabeled sets to evaluate the effectiveness of our semi-supervised method.

Overall, the contributions of this paper are as follows:

- We are the first to explore semi-supervised learning in the class-agnostic motion prediction task.
- We propose MSRM to filter unreliable pseudo labels and re-generate them from reliable ones, which enables the model to learn more from high-quality pseudo labels.
- We introduce two new augmentations, temporal-sampling and BEVMix, for motion prediction, which facilitate consistency regularization in SSL.

Related Work

Class-Agnostic Motion Prediction. Motion prediction aims to predict the future motion of agents based on past observations. Traditional approaches achieve motion prediction through object detection (Zhou and Tuzel 2017; Lang et al. 2019), followed by subsequent trajectory prediction (Chai et al. 2019; Chang et al. 2019; Djuric et al. 2020; Fang et al. 2020). Relying on object detection, however, these approaches may fail to handle unknown object classes (Wu, Chen, and Metaxas 2020). To provide complementary motion information, class-agnostic motion prediction methods avoid dependence on detection and predict motion directly. These methods represent the environment with BEV maps derived from point clouds and aim to predict the 2D displacement vector for each BEV cell along the horizontal plane. MotionNet (Wu, Chen, and Metaxas 2020) and BE-STI (Wei et al. 2022) proposes to perform joint category perception and motion prediction from the BEV maps. LSTM-ED (Schreiber, Hoermann, and Dietmayer 2019) introduce convolutional LSTM (Shi et al. 2015) to aggregate temporal context. Recently, annotation-efficiency methods such as PillarMotion (Luo, Yang, and Yuille 2021) and WeakMotioinNet (Li et al. 2023) have been proposed to train motion prediction models in a self-supervised and weakly-supervised manner, respectively. While (Li et al. 2023) aims to utilize easier-acquired annotations, we explore to make trade-off between performance and the quantity of annotations.

Scene Flow Estimation. Scene flow estimation methods (Liu, Qi, and Guibas 2018; Gu et al. 2019; Sun et al. 2018; Behl et al. 2019; Wang et al. 2021) produce 3D motion fields in a dense manner. In comparison with class-agnostic motion prediction methods that seek to predict the displacements to future from past observations, scene flow estimation aims to estimate the motion flow between two observed point clouds. Nonetheless, estimating dense 3D flow always demands large computation, making it unfeasible for real-time autonomous systems (Li et al. 2023). Moreover, the direct application of scene flow estimation to actual LiDAR point clouds presents inherent challenges, primarily attributed to the lack of consistent one-to-one correspondences (Wang et al. 2022).

Semi-Supervised Learning (SSL). SSL integrates information from limited labeled data and extensive unlabeled data. Consistency-based regularization (Berthelot et al. 2019b; Xie et al. 2020; Laine and Aila 2017; Berthelot et al. 2019a; Tarvainen and Valpola 2017) applies a consistency loss by enforcing invariance on unlabeled data under different augmentations. Pseudo-labeling relies on the model's high confident predictions to produce pseudo-labels (Lee et al. 2013; Bachman, Alsharif, and Precup 2014; Arazo et al. 2020) for unlabeled data and trains them jointly with labeled data. FixMatch (Sohn et al. 2020a) is a combination of both consistency-based regularization and pseudo-labeling approaches. In this study, we adopt the consistency-based self-training paradigm from FixMatch, enhancing the performance of semi-supervised motion prediction through both consistency regularization and pseudo-labeling aspects.

Semi-Supervised Motion Prediction

Problem Formulation

Class-agnostic motion prediction methods take a temporal sequence of LiDAR point cloud frames as input, where all the point clouds are synchronized to the current coordinate system (Wu, Chen, and Metaxas 2020). We denote synchronized point cloud captured at time t as P^t. P^t is then discretized into dense voxels $V^t \in \{0,1\}^{H\times W\times C}$, where 0 indicates the voxel is empty, 1 indicates the voxel is occupied by at least 1 point, and H, W, C are the voxel numbers along X, Y, Z axis respectively. We view C as the feature dimension of an image and V^t as a virtual BEV map with $H \times W$ cells. Then the motion field $M \in \mathbb{R}^{H\times W\times 2}$ in the BEV map is defined as the 2D displacement of each BEV cell to the next timestamp. Taking BEV map sequence $\mathcal{V} = \{V^t\}_{t=1}^{T}$ as input, the motion prediction model aims to predict the motion field M. For SSL, our goal is to train a motion prediction model by leveraging both a large amount of unlabeled data $\mathcal{D}_u = (\mathcal{V}^u)$ and a smaller set of labeled data $\mathcal{D}_l = (\mathcal{V}^l, M^l)$.

SSL for Motion Prediction

For semi-supervised learning, we basically adopt the framework of mean-teacher (Tarvainen and Valpola 2017) for

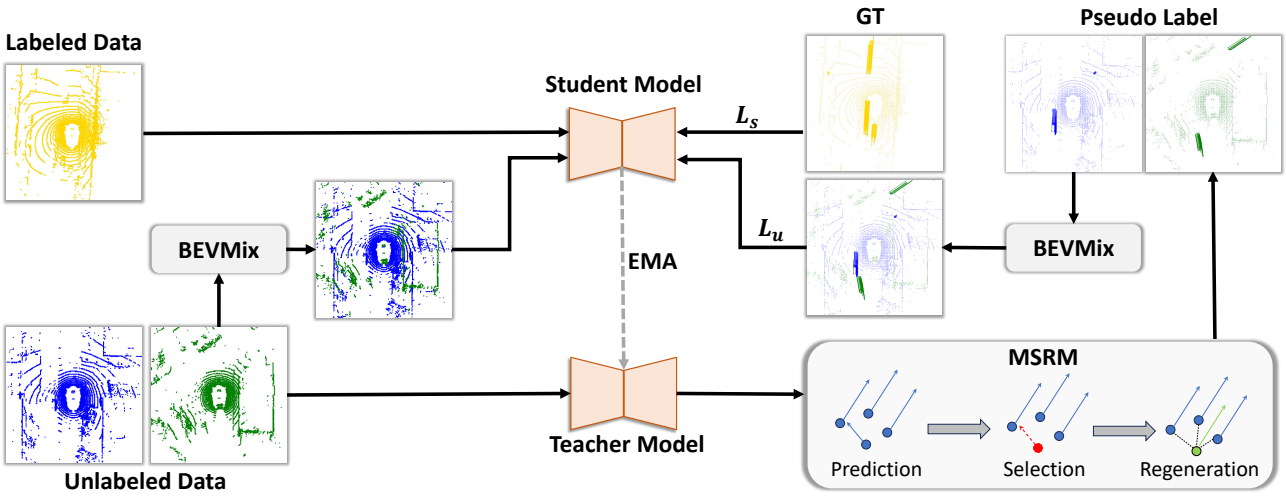

Figure 1: Overview of the proposed approach. Unlabeled samples are first fed to the teacher model to produce pseudo labels, followed by MSRM to improve the quality. Subsequently, the unlabeled samples are mixed using BEVMix and fed to the student model to compute the unlabeled loss. Concurrently, labeled samples are also fed to the student model to compute the supervised loss. The weights of the teacher model are updated from the weights of the student model by Exponential Moving Average (EMA) in every iteration.

its effectiveness and flexibility. This involves two stages of training, where in the first stage, we train a teacher model using all available labeled data, and in the second stage, we train a student model using both labeled and unlabeled data with strong-to-weak consistency. The steps of our semi-supervised training are summarized as follows:

1. Train a teacher model on labeled samples.
2. Generate pseudo labels of unlabeled samples using the trained teacher model and weak augmentations.
3. Select reliable pseudo labels and re-generate the unreliable ones.
4. Apply strong data augmentations to unlabeled samples and train a student model with both the labeled and unlabeled samples.
5. Update teacher model with student model. Repeat steps 2-4.

Training Teacher Model

There are two branches of our pipeline, one teacher model $\mathcal{G}_{\theta^t}$ and one student model $\mathcal{G}_{\theta^s}$. We first train the teacher model on the labeled dataset with supervised loss:

$$\mathcal{L}_s = \ell_{smooth_{l1}}\big(\mathcal{G}_\theta(\mathcal{V}^l), M^l\big). \tag{1}$$

Motion Select and Re-generate Module (MSRM)

We first perform test-time inference using the trained teacher model on unlabeled data to generate pseudo labels $\hat{M}^u$.

Challenge in selecting reliable pseudo labels. However, trained on limited labeled data, the teacher model produces low-quality pseudo labels. Learning from these low-quality labels will negatively impact the performance of the model.

For image classification and object detection, reliable (high-quality) pseudo labels can be selected by classification score or objectiveness score. But for motion prediction, as a regression task, there is no explicit metric to judge the reliability. To this end, we propose the motion select and re-generate module (MSRM), which shows effectiveness in generating reliable pseudo labels for the motion prediction task.

Approach to evaluate the reliability of motion labels. As the pseudo motion labels $\hat{M}^u$ indicate the displacement of each BEV cell from the current frame to the future one, if the motion is accurate, the warped BEV cells would be exactly overlapped with the corresponding future ones. Based on this, we can first warp the current BEV cells with the pseudo label $\hat{M}^u$, and then find the corresponding cells in the future frame. Finally, we can select the reliable pseudo label based on the distance between the corresponding ones. The correspondence can be found by solving the optimal transport problem. We define $B^t = \{b_i^t \in \mathbb{R}^2\}_{i=1}^{N_t}$ as the 2D coordinates of BEV cells on V^t, where N_t is the number of non-empty cells. The warped coordinates are denoted as:

$$\hat{B}^t = B^t + \hat{M}^u \tag{2}$$

Ideally, $\hat{B}^t$ can be matched with $B^{t+1} = \{b_j^{t+1} \in \mathbb{R}^2\}_{j=1}^{N_{t+1}}$ with matching matrix $\pi \in \{0,1\}^{N_t \times N_{t+1}}$:

$$\hat{B}^t = \pi B^{t+1}, \tag{3}$$

To find the optimal matching matrix, we first compute the cost matrix C by the pairwise distances:

$$C_{ij} = 1 - \exp\big(-\frac{||\hat{b}_i^t - b_j^{t+1}||^2}{\theta_c}\big), \tag{4}$$

where θ_c is a temperature parameter. The optimal matching matrix can be approximated by solving an optimal transport

Algorithm 1: Pseudo Label Re-generation

Input: I_R, I_{UR}, B
Output: I^u, M^u

1: $I^u \leftarrow I_R$, $M^u \leftarrow \hat{M}^u(I_R)$
2: **for** $i \in I_{UR}$ **do**
3: $D \leftarrow ||B(i) - B(I_R)||_2$
4: // I_K: index of the K neighbors
5: $D_K, I_K \leftarrow \texttt{TopK}(D)$
6: $D_K, I_K \leftarrow D_K(D_K < \beta), I_K(D_K < \beta)$
7: $M_K \leftarrow M^u(I_K)$
8: **if** $\texttt{Len}(I_K)$ is *zero* **then**
9: continue
10: **end if**
11: // θ_w: temperature parameter
12: $W \leftarrow \exp(-\frac{D_K}{\theta_w})$
13: $M_{mean} \leftarrow \texttt{WeightedMean}(M_K, W)$
14: $M_{dif} \leftarrow \texttt{abs}(\frac{M_K - M_{mean}}{M_{mean}})$
15: $H \leftarrow \exp\left(-\texttt{WeightedMean}(M_{dif}, W)\right)$
16: **if** $H > \gamma$ **then**
17: I^u.add(i), M^u.add(M_{mean})
18: **end if**
19: **end for**

problem (Cuturi 2013):

$$\begin{aligned} \pi^* &= \arg\min_{\pi} \sum_{i,j} C_{ij}\pi_{ij} \\ s.t. \quad \pi \mathbf{1}_{N_{t+1}} &= \frac{1}{N_t}\mathbf{1}_{N_t}, \pi^T \mathbf{1}_{N_t} = \frac{1}{N_{t+1}}\mathbf{1}_{N_{t+1}}, \end{aligned} \tag{5}$$

where π^* is the optimal matching matrix. According to Eq. 2 and Eq. 3, we can obtained auxiliary pseudo labels:

$$\tilde{M}^u = \pi^* B^{t+1} - B^t. \tag{6}$$

Finally, we can evaluate the motion pseudo label's reliability by the difference between $\hat{M}^u$ and $\tilde{M}^u$. The difference is computed by:

$$\Delta M = ||\hat{M}^u - \tilde{M}^u||_2. \tag{7}$$

Select and re-generate pseudo labels. Based on the difference ΔM, we can obtain the indexes for cells with reliable label $I_R = \{i|\Delta M(i) < \mu\}$ and indexes for unreliable ones $I_{UR} = \{i|\Delta M(i) \geq \mu\}$. The key idea is that if the pseudo label is accurate, the cost between the warped cell and its corresponding target cell will be small and the optimal solver is more likely to find the correct correspondence. The process is shown in Fig. 2.

Furthermore, inspired by local rigid characteristics of most objects (*e.g.*, cars) in the driving scenario, we re-generate pseudo labels for unreliable cells from reliable adjacent labels, as shown in Algorithm 1. For each unreliable cell, we first find K nearest reliable neighbor cells based on Euclidean distance (lines 3-5). And the neighbors are valid only if the distances are within the distance threshold β (lines 6-7). Then we use these reliable labels to re-generate

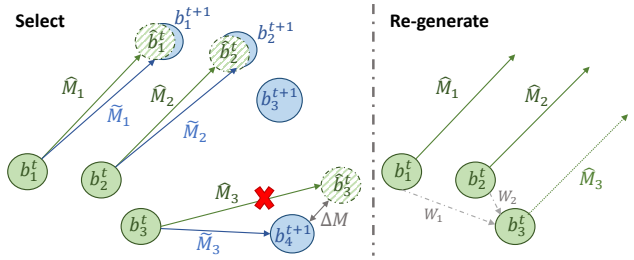

Figure 2: Diagram of MSRM. Select: In the case of an inaccurate pseudo label $\hat{M}_3$, where b_3^t is incorrectly matched with a noise b_4^{t+1}, resulting in a large ΔM. Re-generate: the pseudo label of b_3^t is then re-generated by the weighted mean of neighbor reliable labels.

labels for unreliable cells. We first calculate the weights W based on the distance, the closer the distance is, the larger the weight is (line 12). Then we calculate the weighted mean of the reliable neighbors as the re-generated labels (line 13). To evaluate the quality of the re-generated labels, We calculate the weighted differences between the K neighbors and M_{mean} to measure the local consistency H (lines 14-15). The idea is derived from the fact that most objects in the scene are almost rigid and each part of a single one should have similar motion. A larger H indicates that the unreliable cell and its K neighbors are more likely from the same object. By discarding re-generated labels with low consistency H, we obtain the final pseudo labels M^u and correspondent index I^u for the unlabeled sample (line 17). We only compute unsupervised loss for cells within index I^u.

Data Augmentations

As discussed by FixMatch (Sohn et al. 2020a), weak-to-strong consistency is important for the performance of semi-supervised learning. In our framework, we establish consistency regularization through following process: firstly, the teacher model processes weakly augmented unlabeled samples, generating pseudo-labels through test-time inference. Subsequently, the same samples undergo strong augmentation and are fed to the student model for predictions. Finally, the regularization is achieved by enforcing consistency between the predictions from strongly augmented samples and the pseudo-labels derived from weakly augmented samples. While data augmentation has been widely studied in image classification and object detection, it have not yet been thoroughly explored in the class-agnostic motion prediction task. To this end, we introduce two simple yet effective data augmentations called temporal-sampling and BEVMix.

Temporal-sampling. Taking input $\mathcal{V} = \{V^t\}_{t=1}^T$, we can generate artificial sequence $\mathcal{V}_{\mathcal{TS}} = \{V^1, V^3, V^5, ..., V^t\}$ by uniform-sampling along temporal dimension. And correspondent motion ground truth will be $M_{TS} = 2M$. We then pad $\mathcal{V}_{\mathcal{TS}} = \{V^1, ..., V^1, V^3, V^5, ..., V^t\}$ with V^1 to make sure the sequence length is equal to $\mathcal{V}$. From temporal sampling, we simulate a scenario from original input $\mathcal{V}$, in which objects are static at the beginning and suddenly start

to move with double speed. By temporal-sampling, we can compensate for a number of fast-speed samples, which are relatively fewer in comparison to the static and slow ones in the original dataset.

BEVMix. Mix data augmentations aim to generate artifact samples by combining two existing samples. Mixup (Zhang et al. 2018) proposes to enforce smooth label transitions by linearly interpolating between pairs of input samples and their corresponding class labels. Instead of generating unnatural mixed samples like Mixup, CutMix (Yun et al. 2019) proposes directly replacing the image region with a patch from another training image. These strategies are also lifted from 2D to 3D for point cloud recognition (Chen et al. 2020; Zhang et al. 2022). To be not limited to single objects, Mix3D (Nekrasov et al. 2021) proposes to mix two full point cloud scenes directly by combining all the points from two scenes as a single scene for point cloud segmentation. LaserMix (Kong et al. 2023) leverages the spatial prior of point clouds and mix two scenes by LiDAR scans. Although effective, previous data mix approaches mainly focus on single input while the motion prediction methods take sequence input, as it is important to observe the locations of each object throughout the temporal window to make accurate motion prediction. Directly applying data mix strategies like Cutmix may lead to a case in which we cannot find correspondent cells of a moving object in the current frame from previous frames. Additionally, these data mix approaches are mainly designed for classification or segmentation based on some task-specific priors, which are also not suitable for the motion prediction task.

To this end, we propose a simple but effective BEVMix to mix BEV sequences for class-agnostic motion prediction. As depicted in Algorithm 2, we view one sample as foreground and the other as background. We first remove ground points from the "foreground" sample $\mathcal{V}^f$ by a ground segmentation algorithm (Lee, Lim, and Myung 2022) to reduce noise (line 2), obtain the coordinates of non-empty cells $\{B_t^f\}_{t=1}^T$, and then occupy "background" BEV map with B^f (line 5). Meanwhile, the motion pseudo-label map of the background sample is also occupied by M^f (line 7). Finally, we obtain the mixed sample $\{V_t^{mix}\}$ and the correspondent pseudo labels M^{mix}. The effect of the BEVMix is two-fold. For the objects in the "foreground" sample, BEVMix provides more scenes from the "background" sample. By occupying the "background" sample with the "foreground" sample, the temporal correspondent points in the "foreground" are maintained, which is crucial for motion prediction. For the "background" sample, BEVMix acts like sparse CutMix that replaces original cells with ones from the "foreground" sample. And thanks to the ground-removing and sparse distribution of non-empty cells in the BEV map, most of the temporal correspondent points in the "background" can be maintained after BEVMix, which is also beneficial.

Training Student Model

The unsupervised loss is computed by

$$\mathcal{L}_u = \ell_{smooth_{l1}}\big(\mathcal{G}_{\theta^s}(\mathcal{V}^{mix}), M^{mix}\big). \tag{8}$$

Algorithm 2: BEVMix

Input:$\{B_t^f\}_{t=1}^T$, $\{B_t^b\}_{t=1}^T$, M^f, M^b, $\{V_t^f\}_{t=1}^T$, $\{V_t^b\}_{t=1}^T$
Output:$\{V_t^{mix}\}$, M^{mix}

1: $\{B_t^{mix}\} \leftarrow \{B_t^b\}$, $M^{mix} \leftarrow M^b$, $\{V_t^{mix}\} \leftarrow \{V_t^b\}$
2: $\{B_t^f\}_{t=1}^T \leftarrow$ `GroundRemove` $(\{B_t^f\}_{t=1}^T)$
3: **for** $t \in [1, T]$ **do**
4: **for** $b^f \in B_t^f$ **do**
5: $V_t^{mix}(b^f) \leftarrow V_t^f(b^f)$
6: **if** $t = T$ **then**
7: $M^{mix}(b^f) \leftarrow M^l(b^f)$
8: **end if**
9: **end for**
10: **end for**

The overall loss for the student model is:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u. \tag{9}$$

In every iteration, we use the Exponential Moving Average (EMA) to update the weights of the teacher model from the student model:

$$\theta^t = \alpha\theta^t + (1-\alpha)\theta^s \tag{10}$$

Experiments

Settings

Dataset. Following previous work (Wu, Chen, and Metaxas 2020; Luo, Yang, and Yuille 2021; Wang et al. 2022), We evaluate the proposed approach on the large-scale self-driving dataset: nuScenes (Caesar et al. 2019), which contains 850 scenes with annotations. For fair comparisons, we follow MotionNet to use 500 scenes for training, 100 scenes for validation, and 250 scenes for testing.

SSL settings. For SSL settings, we random sample 1%, 5%, and 10% training scenes as the labeled set and use the rest of the training scenes as an unlabeled set to simulate the scenario that large amounts of data are corrected, but only a minority of it has been annotated.

Implementation details. We follow the same pre-process with MotionNet, where the point clouds are cropped in the range of [-32m, 32m]× [-32m, 32m]× [-3m, 2m] and the voxel size is set to 0.25m ×0.25m × 0.4m along XYZ axis. We take the current sweep and the past four sweeps as input ($t = 5$) and transform the past sweeps into the current coordinate system through ego-motion. Following previous works (Luo, Yang, and Yuille 2021; Li et al. 2023), we apply MotionNet as the baseline model. Adam (Kingma and Ba 2017) is used as the optimizer. We implement our model in Pytorch (Paszke et al. 2019) with a single A6000 GPU. We use the teacher model for evaluation following previous SSL works. We use random flip as weak augmentations and proposed temporal sampling and BEVMix as strong augmentations. The parameters K, μ, β, γ, θ_c, θ_w, and α are set to 5, 1, 10, 0.6, 3, 5, and 0.999, respectively.

Evaluation metrics. Following MotionNet, we evaluate the mean and median errors at different speed levels by dividing the grid cells into 3 groups according to the ground truth

Method	Supervision	Static		Speed $\leq$ 5m/s (Slow)		Speed $\geq$ 5m/s (Fast)	
		Mean↓	Median↓	Mean↓	Median↓	Mean↓	Median↓
LSTM-ED	Full.	0.0358	0	0.3551	0.1044	1.5885	1.0003
MotionNet	Full.	0.0256	0	0.2565	0.0962	1.0744	0.7332
BE-STI	Full.	0.0220	0	0.2115	0.0929	0.7511	0.5413
PillarMotion	Self.	0.1620	0.0010	0.6972	0.1758	3.5504	2.0844
WeakMotionNet	Weak.	0.0243	0	0.3316	0.1201	1.6422	1.0319
Ours	Sup-only (1%)	0.0173	0	0.4882	0.1057	3.7242	2.5059
	Semi.(1%)	**0.0153**	**0**	**0.3497**	**0.1020**	**1.9407**	**1.2173**
	Sup-only(5%)	0.0198	0	0.3532	**0.0990**	2.6661	1.6253
	Semi.(5%)	**0.0183**	**0**	**0.3021**	0.0991	**1.4516**	**0.8864**
	Sup-only (10%)	0.0245	0	0.3159	0.0992	1.9097	1.1771
	Semi. (10%)	**0.0218**	**0**	**0.2746**	**0.0996**	**1.2030**	**0.7880**

Table 1: Results of motion prediction methods on the nuScenes dataset. "Full.", "Self.", "Weak.", and "Semi.", refer to fully-supervised, self-supervised, weakly-supervised, and semi-supervised training, respectively.

MT	TS	SR	BM	Static	Slow	Fast
				Mean Errors ↓		
✓				0.0133	0.4218	3.0868
✓	✓			0.0162	0.4594	2.9037
✓	✓	✓		0.0161	0.3946	2.7705
✓	✓		✓	**0.0139**	0.3581	2.1427
✓	✓	✓	✓	0.0153	**0.3497**	**1.9407**

Table 2: Effectiveness of TS, MSRM (SR), and BEVMix (BM) on nuScenes dataset in 1% labeled data setting.

MT	S	R	C	Static	Slow	Fast
				Mean Errors ↓		
✓				0.0162	0.4594	2.9037
✓	✓			0.0163	0.4469	2.8689
✓	✓	✓		0.0179	0.4579	4.0773
✓	✓	✓	✓	**0.0161**	**0.3946**	**2.7705**

Table 3: Effectiveness of each component in MSRM at 1% labeled data setting. MT stands for MeanTeacher; S stands for selection; R stands for Re-generate; C stands for Consistency-evaluation.

speeds: static, slow ($\leq$ 5m/s), and fast ($\geq$ 5m/s). Errors are measured by L_2 distances between the predicted displacements and the ground truth displacements for the next 1s.

Main Results

Table 1 benchmarks results on the nuScenes dataset (Caesar et al. 2019). With 1% labeled data, our SSL method significantly outperforms self-supervised PillarMotion (*e.g.*, 1.9407 vs 3.5504 errors at the fast speed level). With 5% labeled data, our method surpasses weakly supervised WeakMotionNet. And when with 10% labeled data, our SSL method can achieve a small performance gap with the fully-supervised baseline, MotionNet (0.0218 vs 0.0256 errors at static level; 0.2746 vs 0.2565 errors at slow level; 1.2030 vs 1.0744 errors at fast level). The results indicate that our method can achieve a good trade-off between annotations and performances. Additionally, Table 1 also demonstrates the performance improvements gained from unlabeled data using our proposed method. For instance, with our method, the mean errors at fast speed are reduced from 3.7242 to 1.9407 with 1% labeled data and 99% unlabeled data; from 2.6661 to 1.4516 with 5% labeled data and 95% unlabeled data; from 1.9097 to 1.2030 with 10% labeled data and 90% unlabeled data. The qualitative results are shown in Fig. 3

Ablation Study

Ablation study for the entire framework. In Table 2, we show the performance with different combinations of Temporal-sampling (TS), MSRM, and BEVMix in the 1% labeled data setting. Compared to the SSL baseline, Mean-Teacher, TS reduces the mean error by 0.1831 at the fast level, while increasing the error by 0.0376 at the slow level (compare row 1 with row 2). This indicates that TS successfully produces more fast samples for the model to learn. However, it also makes the samples harder to learn. With MSRM, errors are reduced from 0.4594 to 0.3946 at the slow speed level and from 2.9037 to 2.7705 at the fast speed level (compare row 2 with row 3). BEVMix significantly reduces the error from 0.3946 to 0.3497 at the slow level and from 2.7705 to 1.9407 at the fast level (compare row 3 with row 5), demonstrating that BEVMix is an extremely effective data augmentation.

Ablation study for the MSRM. The effectiveness of each process in MSRM is shown in Table 3. With the only selection, the performance improves slightly (*e.g.*, mean error drop from 2.9037 to 2.8689 at the fast level), demonstrating the effectiveness of discarding unreliable pseudo labels during semi-supervised training. When regenerating pseudo labels without considering local consistency, the prediction results become even worse (*e.g.*, the mean error increases from 2.8689 to 4.0773), which indicates that directly gener-

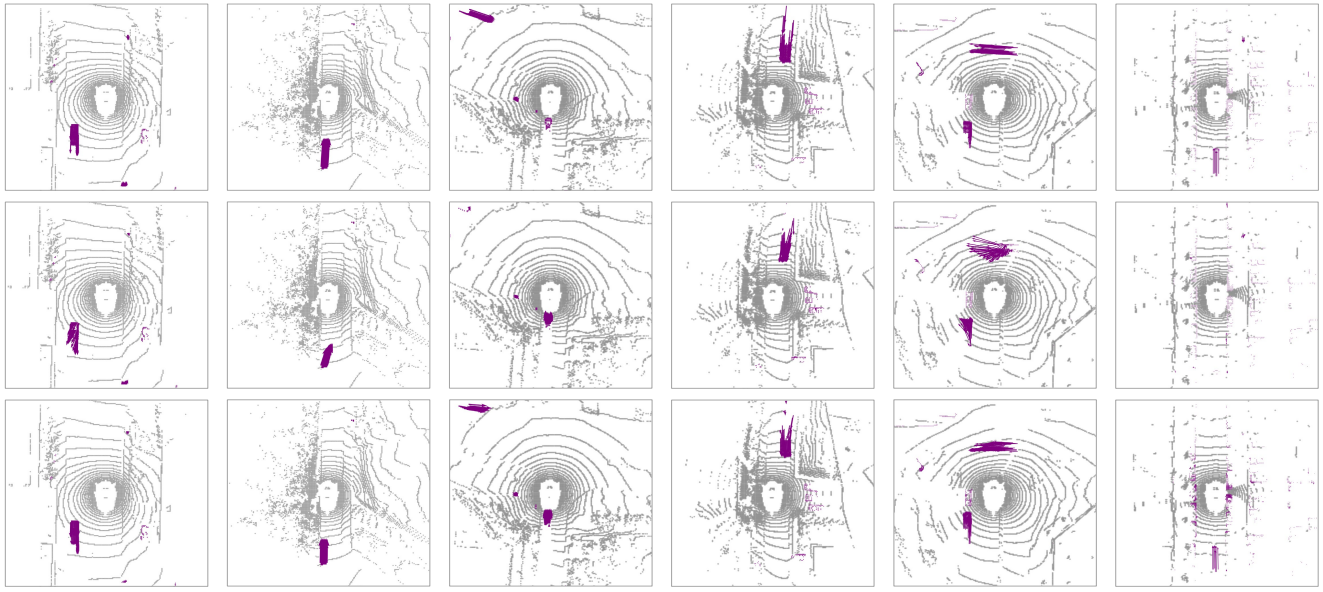

Figure 3: Qualitative results on the nuScenes dataset. First row: ground truth. Second row: results of training with 1% labeled data. Third row: results of our semi-supervised method trained with 1% labeled data and 99% unlabeled data.

Method	Static	Slow	Fast
	Mean Errors ↓		
Mean-Teacher	0.0162	0.4594	2.9037
+ Mixup	**0.0106**	0.4282	2.8902
+ CutMix	0.0166	0.4510	2.6157
+ BEVMix	0.0139	**0.3581**	**2.1427**

Table 4: Effectiveness of different mix strategies in 1% labeled data setting.

Method	Static	Slow	Fast
	Mean Errors ↓		
MotionNet†	0.0262	0.2467	0.9878
+ TS	0.0287	0.2510	0.9568
+ BEVMix	**0.0261**	0.2270	0.8686
+ TS + BEVMix	0.0271	**0.2267**	**0.8427**

Table 5: Effectiveness of TS and BEVMix for fully supervised training. † indicates the results with flip data augmentation.

ating from neighbors produces more low-quality pseudo labels. Finally, by selecting reliable pseudo labels again from the re-generated ones, our model achieve better performance (*e.g.*, error drop from 2.9037 to 2.7705).

Ablation study for the data mix strategy. In Table 4, we compare our proposed BEVMix with the two most widely used data mixing approaches: Mixup and CutMix. Mixup and CutMix can be considered as randomly replacing cells/areas in the current BEV sequence with cells/areas from the other BEV sequence. With Mixup, static errors are reduced by a large margin, but the fast errors only drop slightly. This is because Mixup can effectively maintain correspondence among static cells throughout temporal windows, but it may easily miss corresponding moving cells. CutMix has a considerable improvement over the baseline (*e.g.*, mean error drops from 2.9037 to 2.6157 at the fast level). However, our BEVMix can maintain the moving trajectories better and outperforms the Cutmix by a large margin (2.6157 vs 2.1427 error at the fast speed level), showing BEVMix is a more appropriate data mix approach for the class-agnostic motion prediction.

Apply TS and BEVMix to fully-supervised training. We demonstrate that temporal sampling and BEVMix can also provide benefits in the context of fully supervised learning. As illustrated in Table 5, TS and BEVMix can significantly reduce the fast mean error from 0.9878 to 0.8427.

Conclusion

In this work, we study semi-supervised class-agnostic motion prediction. Specially, we propose two augmentations for the motion prediction task which facilitate the weak-to-strong consistency regularization and significantly improve the performance. Additionally, we propose a novel MSRM module to select and re-generate higher quality pseudo labels, which encourages the model to learn better from the unlabeled data. Experiments show that our semi-supervised method boosts the performance by making use of the unlabeled data and achieving a good trade-off between annotations consumption and performance.

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 also supported by the MoE AcRF Tier 2 grant (MOE-T2EP20220-0007).

References

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Bachman, P.; Alsharif, O.; and Precup, D. 2014. Learning with pseudo-ensembles. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Behl, A.; Paschalidou, D.; Donné, S.; and Geiger, A. 2019. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2019a. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019b. Mixmatch: A holistic approach to semi-supervised learning. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2019. nuScenes: A Multimodal Dataset for Autonomous Driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Chai, Y.; Sapp, B.; Bansal, M.; and Anguelov, D. 2019. MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. arXiv:1910.05449.

Chang, M.-F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. 2019. Argoverse: 3d tracking and forecasting with rich maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Chen, Y.; Hu, V. T.; Gavves, E.; Mensink, T.; Mettes, P.; Yang, P.; and Snoek, C. G. 2020. Pointmixup: Augmentation for point clouds. In *European Conference on Computer Vision (ECCV)*, 330–345. Springer.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Djuric, N.; Radosavljevic, V.; Cui, H.; Nguyen, T.; Chou, F.-C.; Lin, T.-H.; Singh, N.; and Schneider, J. 2020. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

Fang, L.; Jiang, Q.; Shi, J.; and Zhou, B. 2020. Tpnet: Trajectory proposal network for motion prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Kong, L.; Ren, J.; Pan, L.; and Liu, Z. 2023. Lasermix for semi-supervised lidar semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 21705–21715. IEEE.

Laine, S.; and Aila, T. 2017. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Lee, S.; Lim, H.; and Myung, H. 2022. Patchwork++: Fast and Robust Ground Segmentation Solving Partial Under-Segmentation Using 3D Point Cloud. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

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

Liang, M.; Yang, B.; Zeng, W.; Chen, Y.; Hu, R.; Casas, S.; and Urtasun, R. 2020. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Liu, X.; Qi, C.; and Guibas, L. J. 2018. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Luo, C.; Yang, X.; and Yuille, A. L. 2021. Self-Supervised Pillar Motion Learning for Autonomous Driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Nekrasov, A.; Schult, J.; Litany, O.; Leibe, B.; and Engelmann, F. 2021. Mix3d: Out-of-context data augmentation for 3d scenes. In *2021 International Conference on 3D Vision (3DV)*, 116–125. IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

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

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and chun Woo, W. 2015. Convolutional LSTM Network:

A Machine Learning Approach for Precipitation Nowcasting. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020a. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Sohn, K.; Zhang, Z.; Li, C.-L.; Zhang, H.; Lee, C.-Y.; and Pfister, T. 2020b. A Simple Semi-Supervised Learning Framework for Object Detection. arXiv:2005.04757.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Wang, G.; Wu, X.; Liu, Z.; and Wang, H. 2021. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing (TIP)*, 30: 5168–5181.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Wei, Z.; Qi, X.; Bai, Z.; Wu, G.; Nayak, S. P.; Hao, P.; Barth, M. J.; Liu, Y.; and Oguchi, K. 2022. Spatiotemporal Transformer Attention Network for 3D Voxel Level Joint Segmentation and Motion Prediction in Point Cloud. In *IEEE Intelligent Vehicles Symposium (IV)*. IEEE.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; and Le, Q. 2020. Unsupervised data augmentation for consistency training. In *Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Xu, M.; Zhang, Z.; Hu, H.; Wang, J.; Wang, L.; Wei, F.; Bai, X.; and Liu, Z. 2021. End-to-end semi-supervised object detection with soft teacher. In *IEEE Conference on International Conference on Computer Vision (ICCV)*. IEEE.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412.

Zhang, J.; Chen, L.; Ouyang, B.; Liu, B.; Zhu, J.; Chen, Y.; Meng, Y.; and Wu, D. 2022. Pointcutmix: Regularization strategy for point cloud classification. *Neurocomputing*, 505: 58–67.

Zhou, Q.; Yu, C.; Wang, Z.; Qian, Q.; and Li, H. 2021. Instant-teaching: An end-to-end semi-supervised object detection framework. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Zhou, Y.; and Tuzel, O. 2017. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.