Bootstrap Motion Forecasting With Self-Consistent Constraints

Maosheng Ye\textsuperscript{1,*}, Jiamiao Xu\textsuperscript{2}, Xunnong Xu\textsuperscript{2}, Tengfei Wang\textsuperscript{1}, Tongyi Cao\textsuperscript{2}, Qifeng Chen\textsuperscript{1}
\textsuperscript{1}The Hong Kong University of Science and Technology \textsuperscript{2}DeepRoute.AI

Abstract

We present a novel framework to bootstrap Motion forecasting with Self-consistent Constraints (MISC). The motion forecasting task aims at predicting future trajectories of vehicles by incorporating spatial and temporal information from the past. A key design of MISC is the proposed Dual Consistency Constraints that regularize the predicted trajectories under spatial and temporal perturbation during training. Also, to model the multi-modality in motion forecasting, we design a novel self-ensembling scheme to obtain accurate teacher targets to enforce the self-consistencies with multi-modality supervision. With explicit constraints from multiple teacher targets, we observe a clear improvement in the prediction performance. Extensive experiments on the Argoverse motion forecasting benchmark and Waymo Open Motion dataset show that MISC significantly outperforms the state-of-the-art methods. As the proposed strategies are general and can be easily incorporated into other motion forecasting approaches, we also demonstrate that our proposed scheme consistently improves the prediction performance of several existing methods.

1. Introduction

Motion forecasting has been a crucial task for self-driving vehicles that aims at predicting the future trajectories of agents (e.g., cars, pedestrians) involved in traffic. The predicted trajectories can further help self-driving vehicles plan their future actions and avoid potential accidents. Since the future is not deterministic, motion forecasting is intrinsically a multi-modal problem with substantial uncertainties. This implies that an ideal motion forecasting method should produce a distribution of future trajectories or at least multiple most likely ones.

Due to the inherent uncertainty, motion forecasting remains challenging and unsolved yet. Recently, researchers have proposed different architectures based on various representations to encode the kinematic states and context information from HDMap in order to generate feasible multi-modal trajectories [2, 7, 15, 18, 28, 29, 36, 51, 55, 56, 58]. These methods follow a traditional static training pipeline, where frames of each scenario are split into historical frames (input) and future frames (ground truth) in a fixed pattern. Nevertheless, the prediction task is a streaming task in real-world applications, where the current state will become a historical state as time goes by, and the buffer of the historical state is a queue structure to make successive predicted trajectories. As a result, temporal consistency thus becomes a crucial requirement for the downstream tasks for fault and noise tolerance. To tackle this issue, trajectory stitching is widely applied in traditional planning algorithms [13] to ensure stability along the temporal horizon. However, as the trajectory stitching operation is non-differentiable, it cannot be easily incorporated into learning-based models. Though deep-learning-based models show unprecedented motion prediction performance compared with traditional counterparts, they do not explicitly consider temporal consistency, leading to unstable behaviors in downstream tasks such as planning.

Inspired by these phenomena, we raise a question: can we explicitly enforce consistency when training a deep motion prediction model? On the one hand, the predicted trajectories should be consistent given the successive inputs along the temporal horizon, namely temporal consistency. On the other hand, the predicted trajectories should be stable and robust against small spatial noise or disturbance, namely spatial consistency. In this work, we propose a self-supervised scheme, named as MISC, to enforce consistency constraints in both spatial and temporal domains, namely Dual Consistency Constraints. Dual Consistency Constraints could be viewed as an inner-model consistency and can significantly improve the quality and robustness of motion forecasting, without the need for extra data.

On top of the inner-model consistency, we also exploit the intra-model consistency. Multi-modality is another core characteristic of the motion prediction task. Existing datasets [9, 49] only provide a single ground-truth trajectory for each scenario, which can not satisfy multi-choice situations such as junction scenarios. Most methods adopt the winner-takes-all (WTA) [26] or its variants [4, 35] to alleviate this situation. However, WTA tends to produce confused

\*Work done during an internship at DeepRoute.AI.

8504
predictions when two trajectories are very close. In contrast, our method addresses the multi-modality problem by using more robust teacher targets obtained from self-ensembling, which leverages intra-model consistency. Multiple teacher targets can be viewed as a special kind of intra-model distillation while alleviating the problem of multi-modality. Our contributions are summarized as follows,

- We propose self-consistent constraints in both intra and inner model aspects.
- For the inner-model consistency, Dual Consistency Constraints are proposed to enforce temporal and spatial consistency in our model, which is shown to be a general and effective way to improve the overall performance in motion forecasting.
- For the intra-model consistency constraints, a self-ensembling constraint is explicitly exploited to enforce self-consistency with teacher targets, which provides multi-modality supervision for training.

2. Related Work

Motion Forecasting. Traditional methods [22, 43, 54, 62] for motion forecasting mainly utilize HDMap information for the prior estimation and Kalman filter [23] for motion states prediction. With the recent progress of deep learning on big data, more and more works have been proposed to exploit the potential of data mining in motion forecasting. Early efforts [2, 7, 11, 15, 21, 28, 29, 44, 46, 55, 56, 61] explore different representations, including rasterized image, graph representation, point cloud representation and transformer to generate the features for the task and predict the final output trajectories by regression or post-processing sampling. Most of these works focus on finding more effective and compact ways of feature extraction on the surrounding environment (HDMap information) and agent interactions. Based on these representations, other approaches [6, 33, 46, 56, 57, 58] try to incorporate the prior knowledge with traditional methods, which take the predefined candidate trajectories from sampling or clustering strategies as anchor trajectories. To some extent, these candidate trajectories can provide better guidance and goal coverage for the trajectories regression due to straightforward HDMap encoding. Nevertheless, this extra dependency makes the stability of models highly related to the quality of the trajectory proposals. Goal-guided approaches [16, 18, 17] are therefore introduced to optimize goals in an end-to-end manner, paired with sampling strategies that generate the final trajectory for better coverage.

Consistency Regularization. Consistency regularization has been fully studied in semi-supervised and self-supervised learning. Temporally related works [53, 27, 60] apply pairwise matching to minimize the alignment difference through optical flow or correspondence matching to achieve temporal smoothness. Other works [1, 14, 38, 42, 52, 45] apply consistency constraints to predictions from the same input with different transformations in order to obtain perturbation-invariant representations. [8, 3] reverse the temporal order or mask some information and generate pairwise consistency between these predicted trajectories. [48] introduced consistency by examining the gap between agent-centric and scene-centric settings.

Multi-hypothesis Learning. Motion forecasting task inherently has multi-modality due to the future uncertainties and difficulties in acquiring accurate ground-truth labels. WTA [19, 47] in multi-choice learning and its variants [32, 41] incorporate with better distribution estimation to improve the training convergence, thus allowing more multi-modality. Some anchor-based methods [4, 7, 39, 56] introduce pre-defined anchors based on kinematics or road graph topology to provide guidance. However, these methods only allow one target per training stage. Other methods [4, 18] try to generate multi-target for supervision with heavy handcrafted optimizations. We propose a Teacher-Target-Constraints approach to provide more precise trajectory teacher labels by leveraging the power of self-ensembling [25, 59]. Multiple targets are explicitly provided to each agent to better model the multi-modality.

3. Approach

The overall architecture of MISC comprises three parts.

1) We first utilize a joint spatial and temporal learning framework TPCN [55] to extract pointwise features. Based on these features, we decouple the trajectory prediction problem as a two-stage regression task. The first stage performs goal prediction and completes the trajectory with the goal position guidance. The second stage takes the output of the first stage as anchor trajectories for refinement.

2) To enhance the spatial and temporal consistency of our MISC, we introduce Dual Consistency Constraints at the inner-model level, which helps regularize the predictions in a streaming task view.

3) We leverage self-ensembling to generate more precise teacher targets to provide intra-model level self-consistent Teacher Targets Constraints in Sec. 3.3

3.1. Architecture

Recently, TPCN [55] has gained popularity in this task due to its flexibility for joint spatial-temporal learning and scalability to adopt more techniques from point cloud learning. Considering its limitation in representing future uncertainty, we extend TPCN in a two-stage manner through goal position prediction for more accurate waypoints prediction as our baseline. The pipeline is shown in Fig. 1.
**Feature Extraction:** TPCN utilizes dual-representation point cloud learning techniques with multi-interval temporal learning to model the spatial and temporal relationship. All the historical trajectories of input agents and map information are based on pointwise representation \( \{ p_1, p_2, \ldots, p_N \} \), where \( p_i \) is the \( i \)-th point with \( N \) points in total, and then go through multi-representation learning framework to generate pointwise features \( \mathcal{P} \in \mathbb{R}^{N \times C} \), where \( C \) is the channel number.

**Goal Prediction:** With the pointwise features from the backbone, we also adopt the popular goal-based ideas [16, 18, 58] to find the optimal planning policy. Specifically, we first gather all corresponding pointwise agent features and then sum over features to get the agent instance feature \( \phi \in \mathbb{R}^{1 \times C} \). To generate \( K \) goal position prediction \( G = \{ G^k : (g_{x}^{k}, g_{y}^{k}) | 1 \leq k \leq K \} \), we use a simple MLP layer: \( G = \text{MLP}(\phi) \). Instead of relying on heavy sampling strategies like previous goal-based methods, our method avoids generating extra proposals, which may lead to a large computation overhead.

**Trajectory Completion:** With the predicted goal positions, we need to complete each trajectory conditioned on these goals. We propose a simple trajectory completion module to generate \( K \) full trajectories \( \{ \tau_{reg}^k | 1 \leq k \leq K \} \) with a single MLP layer as follows:

\[
\tau_{reg}^k = \{(x_{1}^{k}, y_{1}^{k}), (x_{2}^{k}, y_{2}^{k}), \ldots, (x_{T}, y_{T})\},
\]

\[
\tau_{reg}^k = \text{MLP}(\text{concat}(\phi, G^k)).
\]

**Trajectory Refinement:** Inspired by Faster-RCNN [40] and Cascade-RCNN [5], we use the output trajectories from the Trajectory Completion as anchor trajectories to refine trajectories and predict the corresponding possibility of each trajectory. In particular, the input of the trajectory refinement module will be the whole trajectory with agent historical waypoints \( \tau_{\text{history}} \). With a residual block followed by a linear layer \( \text{Reg} \) and \( \text{Cls} \) respectively, we regress the delta offset to the first stage outputs \( \Delta_{\text{reg}} = \text{Reg}(\tau_{\text{reg}}, \tau_{\text{history}}) \) and corresponding scores \( \tau_{\text{cls}} = \{ e^k | 1 \leq k \leq K \} \) respectively, where \( \tau_{\text{cls}} = \text{Cls}(\tau_{\text{reg}}, \tau_{\text{history}}) \). The final output trajectories will be \( \tau_{\text{reg}}' = \Delta_{\text{reg}} + \tau_{\text{reg}} \).

### 3.2. Dual Consistency Constraints

Consistency regularization has been proved as an effective self-constraint that improves robustness against disturbances. We thus propose inner-model level Dual Consistency Constraints in both spatial and temporal domains to align predicted trajectories for continuity and stability.

#### 3.2.1 Temporal Consistency

In motion forecasting, since each scenario contains multiple successive frames within a fixed temporal chunk, it is reasonable to assume that any two overlapping chunks of input data with a small time shift should produce consistent results. The motion forecasting task aims to predict \( K \) possible trajectories with \( T \) time steps for one scenario, given \( M \) frames historical information. Suppose the information at each history frame is \( I_i \), where \( 1 \leq i \leq M \) and the \( k \)-th output future trajectories are \( \{ (x_{i}^{k}, y_{i}^{k}) | M \leq i < M + T \} \). We first apply time step shift \( s \) for the input for temporal consistency. Therefore, the input history frames information will be \( \{ I_i | 1 + s \leq i \leq M + s \} \) and then we apply the same network for the shifted history information with surrounding HDMap information to generate the \( k \)-th output trajectories \( \{ (x_{i}^{k}, y_{i}^{k}) | \text{M} + s < i \leq \text{M} + s + T \} \). When \( s \) is small, the driving intentions or behavior keeps stable for a short period. Since both trajectories have \( T - s \) overlapping waypoints, they should be as close as possible and share consensus. Thus, we can construct self-constraints for a single scenario input due to the streaming property of the input data. Fig. 2 demonstrates the overall idea of the temporal consistency constraint.
**Trajectory Matching:** Since we predict \( K \) future trajectories to deal with the multi-modality, it is crucial to consider the trajectory matching relationship between original predictions and time-shifted predictions when applying the temporal consistency alignment. For a matching problem, the metric on similarity criteria and matching strategies will be two key factors. Several ways can be used to measure the difference between trajectories, such as Average Displacement Error (ADE) and Final Displacement Error (FDE). We utilize FDE as the criteria since the last position error can partially reflect the similarity with less bias from averaging compared with ADE.

**Matching Strategy:** There are roughly four ways used for matching, namely forward matching, backward matching, bidirectional matching, and Hungarian matching. Forward matching takes one trajectory in the current frame and finds its corresponding trajectory in the next frame with the least cost or maximum similarity. Backward matching is the reverse way compared to forward matching. Further, bidirectional matching consists of both forward and backward matching, which considers the dual relationship. Hungarian matching is a linear optimal matching solution based on linear assignment. Forward and backward matching only considers the one-way situation, which is sensitive to noise and unstable. Hungarian matching has a high requirement for cost function choice. Based on these observations, we choose bidirectional matching as our strategy. We also show its advantages over the other approaches in Sec. 4.3. After obtaining the optimal matching pairs \( \{(m_k, n_k)|1 \leq k \leq K\} \), we can compute the consistency constraint by a simple smooth \( L_1 \) loss [40] \( \mathcal{L}_{Huber} \):

\[
\mathcal{L}_{temp} = \sum_{k=1}^{K} \sum_{t=s+1}^{T} \mathcal{L}_{Huber}(x_{t}^{m_k}, y_{t}^{n_k}; (x'_{t-\Delta}, y'_{t-\Delta})).
\]

(3)

### 3.2.2 Spatial Consistency

Since our MISC is a two-stage framework, the second stage mainly aims for trajectory refinement. It will be more convenient to add spatial permutation in the second stage with less computational cost. First, we apply spatial permutation function \( Z \), including flipping and random noise, to the trajectories from the first stage. The refinement module will process these augmented inputs and generate the offset to the ground truth and classification scores. Under the small spatial permutation and disturbance, we assume that the outputs of the network should also be self-consistent, meaning that the outputs have strong stability or tolerance to noise. Compared with data augmentation, it is explicit regularization. Then the spatial consistency constraint \( \mathcal{L}_{spa} \) is as follows:

\[
\mathcal{L}_{spa} = \mathcal{L}_{Huber}(\Delta_{\tau_{reg}}, Z^{-1}(Reg(Z(\tau_{reg}, \tau_{history}))).
\]

Then the total loss for Dual Consistency Constraints module will be \( \mathcal{L}_{cons} = \mathcal{L}_{spa} + \mathcal{L}_{temp} \).

### 3.3. Teacher-Target Constraints

Teacher-Target Constraints enforce intra-model consistency that not only leverages the power of knowledge distillation but also helps alleviate the multi-modality supervision problem. Existing datasets [9, 49] only provide a single ground-truth trajectory for the target agent, which is to be predicted in one scenario. In order to encourage the multi-modality of models, the winner-takes-all (WTA) strategy is commonly used to prevent the model from collapsing into a single domain. However, the WTA training strategy suffers from instability associated with network initialization. Some other approaches [4, 35] introduce robust estimation methods to select better hypotheses. To some extent, these methods can only implicitly model the multi-modality. Some other approaches [4, 58] generate several possible future trajectories based on the kinematics model and road graph topology. DenseTNT [18]
only uses teacher labels to predict goal sets through a hill-climbing algorithm. These optimization methods tend to impose strict constraints and handcrafted prior knowledge, resulting in inaccurate teacher-targets and inferior performance. In contrast, our approach aims to generate more accurate teacher targets to provide explicit multi-modality supervision through self-ensembling to leverage the power of semi-supervised learning and knowledge distillation.

**Teacher-Target Generation.** The key part of our approach lies in generating more accurate teacher labels for each agent. However, it is straightforward to apply model ensembling techniques [20, 24, 50] to obtain more powerful predictions. Compared with previous works [4, 7, 58], we do not rely on handcrafted anchor trajectory sampling, which is based on inaccurate prior knowledge, including motion estimation. Meanwhile, soft targets from ensembling can better fine-tune the predictions and reduce the gradient variance for better training convergence. As suggested in works [10, 37], the prediction error decreases when the ensemble approach is used once the model is diverse enough. Therefore, we apply k-means algorithm [31] to the predicted trajectories that are collected within different training procedures (for example, launched with different seeds of random number generators, optimized with different learning rates, etc.) of MISC without Teacher-Target Constraints to generate J trajectories with corresponding scores for each scenario. Fig. 3 shows the overall process of our approach. Then with the original ground-truth label, we will formulate J + 1 target trajectories as follows:

\[
\tau_{\text{conf}} = \{c_0, c_1, \ldots, c_J\}, \tag{5}
\]

\[
\tau_{\text{tgt}}^j = \{ (x_1^{tgt_j}, y_1^{tgt_j}), (x_2^{tgt_j}, y_2^{tgt_j}), \ldots, (x_T^{tgt_j}, y_T^{tgt_j}) \}, \tag{6}
\]

where \(\tau_{\text{tgt}}^j\) is the \(j\)-th trajectory with score \(c_j\), among \(J + 1\) target trajectories. To simplify the notation, \(\tau_{\text{tgt}}^0\) is the ground-truth trajectory with \(c_0\) set to 1.

### 3.4. Learning

The total supervision of our MISC can be decoupled into several parts, as described in previous sections. For the regression and classification parts, we loop over all the possible \(J + 1\) targets \(\tau_{\text{tgt}}\). For each target \(\tau_{\text{tgt}}^j\) with confidence \(\tau_{\text{conf}}^j\), we apply WTA strategy as described in Sec. 3.3. Suppose \(K\)-th trajectory from trajectory refinement output \(\tau_{\text{reg}}^j\) is the best trajectory which has the maximum similarity with target \(\tau_{\text{tgt}}^j\), the classification loss and regression loss are defined as:

\[
L_{\text{cls}} = \frac{1}{K} \sum_{k=1}^{K} \tau_{\text{conf}}^j \mathcal{L}_{\text{Huber}}(c_k^j, c_k^*), \tag{7}
\]

\[
L_{\text{reg}} = \frac{1}{T} \sum_{t=1}^{T} \tau_{\text{conf}}^j \mathcal{L}_{\text{Huber}}((x_t^{tgt_j}, y_t^{tgt_j}), (x_t^j, y_t^j)). \tag{8}
\]

For classification loss design, we adopt the displacement prediction idea from TPCN [55] to alleviate the hard assignment phenomenon. As for converting the displacement into probability, we use the standard softmax function to distribute the scores. Since we have trajectory completion and refinement modules, the regression loss will be

\[
L_{\text{reg}} = \sum_{j=0}^{J} (L_{\text{reg}}^j + L_{\Delta\text{reg}}^j), \tag{9}
\]

where \(L_{\Delta\text{reg}}^j\) is the regression loss for the refinement module. The final loss is

\[
L = L_{\text{reg}} + L_{\text{cls}} + L_{\text{cons}}.
\]

### 4. Experiments

We conduct experiments on the Argoverse dataset [9], one of the largest publicly available motion forecasting datasets. We compare our MISC with other state-of-the-art methods. Furthermore, we provide ablation studies to evaluate the effectiveness and generalization ability of each proposed module and design experiments for some hyperparameter choices.

#### 4.1. Experimental Setup

**Dataset.** Argoverse [9] provides more than 300K scenarios with rich HDMap information. For each scenario, objects are divided into three types: agent, AV and others, where “agent” is the object to be predicted. Moreover, each scenario contains 50 frames sampled at 10 Hz, meaning that the time interval between successive frames is 0.1s. The whole dataset is split into training, validation, and test sets, with 205, 942, 39, 472, and 78, 143 sequences, respectively. Waymo open motion dataset (WOD) contains multiple types of agents including vehicles, pedestrians, and cyclists. A total of more than 100,000 segments are provided with more than 1500 km of roadway coverage.

**Metrics.** We use the standard evaluation metrics, including ADE and FDE. ADE is defined as the average displacement error between ground-truth trajectories and predicted trajectories over all time steps. FDE is defined as displacement error between ground-truth trajectories and predicted trajectories at the last time step. We predict \(K\) candidate trajectories for each scenario and calculate the metrics with the ground truth labels. Accordingly, minADE and minFDE are minimum ADE and FDE over the top \(K\) predictions. Moreover, miss rate (MR) is also considered, defined as the percentage of the best-predicted trajectories whose FDE is within a threshold (2m). Brier-minFDE is the minFDE plus
(1 − \( p \))^2$, where \( p \) is the corresponding trajectory probability. Metrics for \( K = 1 \) and \( K = 6 \) are used in our experiments. Note that Brier-minFDE\(_6\) is the ranking metric.

**Experimental Details.** We apply some data augmentation, including random flipping with a probability of 0.5 and global random scaling with a scaling ratio between [0.8, 1.25] during the training stage. As for model settings, the time shift \( s \) for the temporal consistency constraint is set to 1. We adopt \( K = 6 \) to generate 6 trajectories and use \( J = 6 \) teacher targets for each scenario. Furthermore, we choose bidirectional matching for temporal consistency constraint. We finally use 10 models for ensembling due to computation resource limits. For more training details, we have included them in the supplementary materials.

### 4.2. Experimental Results

#### 4.2.1 Results on Argoverse Dataset

**Argoverse Leaderboard Results.** We provide detailed quantitative results of our MISC on the Argoverse test set as well as public state-of-the-art methods in Tab. 1. Compared with previous methods, our MISC improves all the evaluation metrics except MR\(_6\) by a large margin. Further, since the proposed modules are all general training components, other existing motion forecasting models can also benefit greatly from these strategies.

**Qualitative Results.** We also present some qualitative results on the Argoverse validation set in Fig. 4. Compared with results without consistency, the Dual Consistency Constraints improve both the quality and smoothness of the predicted trajectories significantly, resulting in more feasible and stable results despite the input noise.

#### 4.2.2 Results on Waymo Open Motion Dataset

We provide some quantitative results on the validation set of the Waymo Open dataset motion prediction task [12], shown in Tab. 2. Compared with KEMP [30] and Scene-Transformer [36], we also achieve very promising results and show comparable improvement, demonstrating the effectiveness of our approach. We also provide some ablation studies on WOD in the supplementary materials.

### 4.3. Ablation Studies

**Component Study.** As shown in Tab. 3, we conduct an ablation study for our MISC on the Argoverse validation set to evaluate the effectiveness of each proposed component. We adopt TPCN [55] as the baseline shown in the first row of Tab. 3 and add the proposed components progressively. The architecture modifications from the goal set prediction and trajectory refinement module show their promising improvements of about 2%. Dual consistency Constraints have the largest improvements of more than 5% among all the evaluation metrics. Especially for minFDE\(_1\), temporal consistency can optimize 20 cm, indicating the temporal constraints can improve both final position and trajectory probability prediction. Compared with temporal consistency, spatial consistency has less effect on models since we only enforce this constraint in the trajectory refinement stage. Finally, the Teacher-Target Constraints significantly increase performance, manifesting its effectiveness in helping training convergence.

**Temporal Consistency Factors.** We study the factors in the matching problems, including similarity and matching strategies. As shown in Tab. 4, both Hungarian and Bidirectional matching show their advantages over the single
<table>
<thead>
<tr>
<th>Models</th>
<th>minADE</th>
<th>minFDE</th>
<th>MR</th>
<th>minADE</th>
<th>minFDE</th>
<th>MR</th>
<th>b-FDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jean [9, 34]</td>
<td>1.74</td>
<td>4.24</td>
<td>0.68</td>
<td>0.98</td>
<td>1.42</td>
<td>0.13</td>
<td>2.12</td>
</tr>
<tr>
<td>LaneConv [28]</td>
<td>1.71</td>
<td>3.78</td>
<td>0.59</td>
<td>0.87</td>
<td>1.36</td>
<td>0.16</td>
<td>2.05</td>
</tr>
<tr>
<td>LaneRCNN [56]</td>
<td>1.68</td>
<td>3.69</td>
<td>0.57</td>
<td>0.90</td>
<td>1.45</td>
<td>0.12</td>
<td>2.15</td>
</tr>
<tr>
<td>mmTransformer [29]</td>
<td>1.77</td>
<td>4.00</td>
<td>0.62</td>
<td>0.87</td>
<td>1.34</td>
<td>0.15</td>
<td>2.03</td>
</tr>
<tr>
<td>SceneTransformer [36]</td>
<td>1.81</td>
<td>4.06</td>
<td>0.59</td>
<td>0.80</td>
<td>1.23</td>
<td>0.126</td>
<td>1.88</td>
</tr>
<tr>
<td>TNT [58]</td>
<td>1.77</td>
<td>3.91</td>
<td>0.59</td>
<td>0.94</td>
<td>1.54</td>
<td>0.13</td>
<td>2.14</td>
</tr>
<tr>
<td>DenseTNT [18]</td>
<td>1.68</td>
<td>3.63</td>
<td>0.58</td>
<td>0.88</td>
<td>1.28</td>
<td>0.125</td>
<td>1.97</td>
</tr>
<tr>
<td>PRIME [46]</td>
<td>1.91</td>
<td>3.82</td>
<td>0.59</td>
<td>1.22</td>
<td>1.55</td>
<td>0.12</td>
<td>2.09</td>
</tr>
<tr>
<td>TPCN [55]</td>
<td>1.58</td>
<td>3.49</td>
<td>0.56</td>
<td>0.88</td>
<td>1.24</td>
<td>0.13</td>
<td>1.92</td>
</tr>
<tr>
<td>HOME [16]</td>
<td>1.70</td>
<td>3.68</td>
<td>0.57</td>
<td>0.89</td>
<td>1.29</td>
<td>0.08</td>
<td>1.86</td>
</tr>
<tr>
<td>MultiPath++ [51]</td>
<td>1.623</td>
<td>3.614</td>
<td>0.564</td>
<td>0.790</td>
<td>1.214</td>
<td>0.13</td>
<td>1.793</td>
</tr>
<tr>
<td>Hivt++ [61]</td>
<td>1.56</td>
<td>3.44</td>
<td>0.563</td>
<td>0.767</td>
<td>1.146</td>
<td>0.12</td>
<td>1.817</td>
</tr>
<tr>
<td>Ours</td>
<td>1.476</td>
<td>3.251</td>
<td>0.532</td>
<td>0.766</td>
<td>1.135</td>
<td>0.11</td>
<td>1.756</td>
</tr>
</tbody>
</table>

Table 1. The detailed results of our MISC and other top-performing approaches on the Argoverse test set. And b-FDE<sub>6</sub> is the abbreviation of brier-minFDE<sub>6</sub>.

Figure 4. The past trajectory is in yellow, the predicted trajectory is in green, and the ground truth is in red. The top row of the figure shows the results without consistency, while the bottom row shows the results with consistency.

<table>
<thead>
<tr>
<th>Method</th>
<th>minADE</th>
<th>minFDE</th>
<th>Miss Rate</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline  [12]</td>
<td>0.675</td>
<td>1.349</td>
<td>0.183</td>
<td>0.268</td>
</tr>
<tr>
<td>KEMP [30]</td>
<td>0.5691</td>
<td>1.1993</td>
<td>0.1458</td>
<td>0.394</td>
</tr>
<tr>
<td>SceneTrans [36]</td>
<td>0.613</td>
<td>1.22</td>
<td>0.157</td>
<td>0.284</td>
</tr>
<tr>
<td>Ours</td>
<td>0.54</td>
<td>1.11</td>
<td>0.128</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Table 2. Quantitative results on the validation set of the Waymo Open dataset motion prediction task.

direction matching. Although Hungarian matching can ensure the one-to-one matching relationship, it is sensitive to the similarity metric and numerical precision, both of which are not stable in the early training stage. In contrast, bidirectional matching with the FDE similarity metric nearly achieves the best results across all the evaluation metrics. Meanwhile, we also conduct experiments to find the best time-shift value <i>s</i> in the temporal consistency. The details can be found in the supplement.

**Reduction on Temporal Inconsistency.** We use the average L2 distance among all predicted trajectory waypoints to measure the temporal consistency. As shown in Fig. 5, our model without temporal consistency will have large incon-
### Table 3. Ablation study results of modules. Goal refers to Trajectory completion with goal prediction. “Ref.” is the trajectory refinement module, and the “Temp.” is temporal consistency. TTC refers to Teacher-Target Constraints during training.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Consistency</th>
<th>TTC</th>
<th>K=1</th>
<th>K=6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Temp. Spatial</td>
<td>minADE</td>
<td>minFDE</td>
<td>minADE</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.34 2.95</td>
<td>0.73 1.13</td>
<td>1.26 2.77</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.33 2.91</td>
<td>0.725 1.10</td>
<td>1.22 2.67</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.31 2.89</td>
<td>0.71 1.07</td>
<td>1.24 2.70</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.24 2.70</td>
<td>0.705 1.05</td>
<td>1.26 2.77</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.22 2.67</td>
<td>0.70 1.05</td>
<td>1.28 2.80</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.21 2.60</td>
<td>0.67 0.99</td>
<td>1.29 2.80</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.20 2.50</td>
<td>0.66 0.98</td>
<td>1.30 2.80</td>
</tr>
</tbody>
</table>

### Table 4. Ablation study on matching factor for temporal consistency. In this experiment, we remove the Teacher-Target Constraints to fairly study the effect.

<table>
<thead>
<tr>
<th>Matching Strategy</th>
<th>Similarity</th>
<th>minADE</th>
<th>minFDE</th>
<th>MR</th>
<th>minADE</th>
<th>minFDE</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward</td>
<td>ADE</td>
<td>1.25</td>
<td>2.70</td>
<td>0.46</td>
<td>0.670</td>
<td>0.982</td>
<td>0.089</td>
</tr>
<tr>
<td></td>
<td>FDE</td>
<td>1.24</td>
<td>2.68</td>
<td>0.46</td>
<td>0.667</td>
<td>0.958</td>
<td>0.085</td>
</tr>
<tr>
<td>Backward</td>
<td>ADE</td>
<td>1.22</td>
<td>2.67</td>
<td>0.46</td>
<td>0.666</td>
<td>0.972</td>
<td>0.087</td>
</tr>
<tr>
<td></td>
<td>FDE</td>
<td>1.22</td>
<td>2.67</td>
<td>0.46</td>
<td>0.653</td>
<td>0.954</td>
<td>0.084</td>
</tr>
<tr>
<td>Bidirectional</td>
<td>ADE</td>
<td>1.24</td>
<td>2.69</td>
<td>0.45</td>
<td>0.660</td>
<td>0.968</td>
<td>0.088</td>
</tr>
<tr>
<td></td>
<td>FDE</td>
<td>1.23</td>
<td>2.69</td>
<td>0.45</td>
<td>0.660</td>
<td>0.968</td>
<td>0.088</td>
</tr>
<tr>
<td>Hungarian</td>
<td>ADE</td>
<td>1.24</td>
<td>2.69</td>
<td>0.45</td>
<td>0.660</td>
<td>0.968</td>
<td>0.088</td>
</tr>
<tr>
<td></td>
<td>FDE</td>
<td>1.23</td>
<td>2.69</td>
<td>0.45</td>
<td>0.660</td>
<td>0.968</td>
<td>0.088</td>
</tr>
</tbody>
</table>

### Table 5. Ablation study results on the teacher target number \( J \)

<table>
<thead>
<tr>
<th>Teacher Target Num ( J )</th>
<th>K=1</th>
<th>K=6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>minADE</td>
<td>minFDE</td>
</tr>
<tr>
<td>1</td>
<td>1.29</td>
<td>2.82</td>
</tr>
<tr>
<td>3</td>
<td>1.28</td>
<td>2.80</td>
</tr>
<tr>
<td>6</td>
<td>1.26</td>
<td>2.77</td>
</tr>
</tbody>
</table>

### Table 6. Ablation study of consistency constraints and Teacher Target Constraints on different state-of-the-art methods on Argoverse validation set. Performance for methods without constraints is obtained from corresponding papers or our reproduction.

<table>
<thead>
<tr>
<th>Method</th>
<th>Consistency</th>
<th>TTC</th>
<th>K=1</th>
<th>K=6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Temp. Spatial</td>
<td>minADE</td>
<td>minFDE</td>
<td>minADE</td>
</tr>
<tr>
<td>LaneGCN [28]</td>
<td>× ×</td>
<td>1.35</td>
<td>2.97</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.29</td>
<td>2.80</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.30</td>
<td>2.88</td>
<td>0.69</td>
</tr>
<tr>
<td>TPCN [55]</td>
<td>× ×</td>
<td>1.34</td>
<td>2.95</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.27</td>
<td>2.79</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.30</td>
<td>2.86</td>
<td>0.69</td>
</tr>
<tr>
<td>mmTransformer [29]</td>
<td>× ×</td>
<td>1.38</td>
<td>3.03</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.31</td>
<td>2.83</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.29</td>
<td>2.80</td>
<td>0.68</td>
</tr>
<tr>
<td>DenseTNT [18]</td>
<td>× ×</td>
<td>1.36</td>
<td>2.94</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.25</td>
<td>2.81</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td>1.30</td>
<td>2.82</td>
<td>0.69</td>
</tr>
</tbody>
</table>

### 4.4. Generalization Capability

To verify the generalization capability of Dual Consistency Constraints and Teacher Targets Constraints, we also apply them to different models with state-of-the-art performance to show that they can be plugin-in training schemes. **Consistency Component.** As shown in Tab. 6, our dual consistency constraints can effectively improve the performance of models regardless of their representations through the training phase. There is a noticeable improvement of over 5% on every metric, especially for minFDE.

**Teacher Target.** Teacher-Target Constraints is another general training trick that can be widely used in other frameworks. In Tab. 6, we also verify its effectiveness on other public methods. Methods with Teacher-Target Constraints have nearly over 3% improvement in all metrics. For the original DenseTNT [18], we replace its original hand-crafted optimization for teacher goal targets with our self-ensembling teacher targets. This strategy brings an over 5% increase in performance, demonstrating the better quality of the self-ensembling teacher targets than handcrafted optimizations and estimation.

### 5. Conclusion

In this work, we propose MISC, an effective architecture for the motion forecasting task. We impose inner-model dual consistency regularization on both spatial and temporal domains to leverage the potential of self-supervision, which has been ignored by previous efforts. Besides, we explicitly model the multi-modality by providing supervision and constraints with powerful self-ensembling techniques in an intra-model aspect. Experimental results on the Argoverse motion forecasting dataset and Waymo dataset show the effectiveness of our approach and generalization capability to other methods.
References


[28] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representa-


[50] Anni Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems, 30, 2017. 5


