2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# Improving Autonomous Driving Safety with POP: A Framework for Accurate Partially Observed Trajectory Predictions

Sheng Wang, Yingbing Chen, Jie Cheng, Xiaodong Mei, Ren Xin, Yongkang Song and Ming Liu

***Abstract*— Accurate trajectory prediction is crucial for safe and efficient autonomous driving, but handling partial observations presents significant challenges. To address this, we propose a novel trajectory prediction framework called Partial Observations Prediction (POP) for congested urban road scenarios. The framework consists of two key stages: self-supervised learning (SSL) and feature distillation. POP first employs SLL to help the model learn to reconstruct history representations, and then utilizes feature distillation as the fine-tuning task to transfer knowledge from the teacher model, which has been pre-trained with complete observations, to the student model, which has only few observations. POP achieves comparable results to top-performing methods in open-loop experiments and outperforms the baseline method in closed-loop simulations, including safety metrics. Qualitative results illustrate the superiority of POP in providing reasonable and safe trajectory predictions. Demo videos and code are available at https://chantsss.github.io/POP/.**

## I. Introduction

The rapid development of autonomous vehicles has brought a myriad of challenges and opportunities to both academia and industry in recent years. One of the critical aspects of self-driving technology is vehicle trajectory prediction, which provides valuable information for autonomous vehicles to assess potential risks and make informed decisions in dynamic traffic situations. Challenges in this domain include the dynamic and unpredictable nature of traffic, interactions between road users, diverse driving behaviors, sensor occlusions and limitations. Recently, data-driven approaches exhibited promising performance in prediction accuracy on challenges [1] [2]. A motion forecasting model typically collects comprehensive information from perception signals and highdefinition (HD) maps, such as traffic light states, motion history of agents, and the road graph. The most state-of-the-art prediction models adopt Transformers [3] [4] [5] or graph neural networks (GNNs) [6] [7] to encode agent-agent and agent-map interactions have achieved outstanding prediction accuracy. Some researchers proposed to employ Self-supervised learning (SSL) to train a network for more transferable, generalizable, and robust representation learning. For example, SSL-Lanes [8] and PreTraM [9] demonstrated that carefully designed pretext tasks can significantly enhance performance without using extra data by learning richer features.

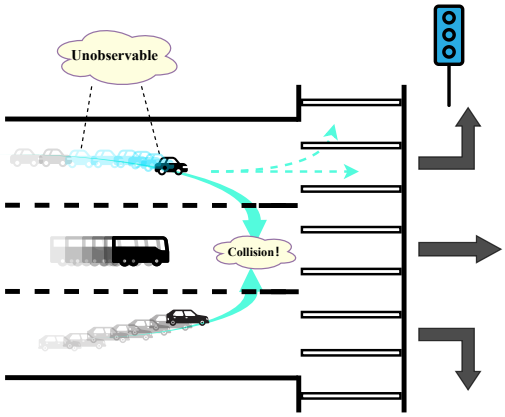


Fig. 1. **Partially observed predictions in real-world situations.** In this scenario, the self-driving car is making a left turn, but another car is accidentally turning right from the left turn lane. Due to insufficient observations, the future trajectories provided by the prediction algorithm fail to include this possibility, leading to a dangerous situation.

However, these methods focus solely on fitting an inference model on a dataset with a neural network, without considering the mismatch between the distribution of the actual noisy data from the upstream module and the clean data provided on the dataset. This mismatch is mainly due to the limitations of the sensing and tracking system equipment and algorithms in the real world, which is known as the sim-2-real problem. Some works have explored how to improve the robustness of predictors from the perspective of input noise [10] [11], but they have neglected a key phenomenon in practical applications, namely, domain shift due to insufficient observation data, which is common to see in autonomous driving scenarios.

In concrete terms, most current learning-based prediction algorithms require a fixed-length history trajectory as output, as per the popular challenge setting. For example, Argoverse 1 (Av1) [1] provides 2 seconds of history information, while Argoverse 2 (Av2) [2] requires 5 seconds of observations as input for longer-term prediction tasks. However, such

This work was supported by Lotus Technology Ltd. through The Hong Kong University of Science and Technology (GZ) under Cooperation Project R00082. (Corresponding author: Ming Liu.)

Sheng Wang, Yingbing Chen and Ren Xin are with Robotics and Autonomous Systems, Division of Emerging Interdisciplinary Areas (EMIA) under Interdisciplinary Programs Office (IPO), The Hong Kong University of Science and Technology, Hong Kong SAR, China. {swangei, ychengz, rxin}@connect.ust.hk

Jie Cheng is with Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China. jchengai@connect.ust.hk

Xiaodong Mei is with Computer Science, The Hong Kong University of Science and Technology, Hong Kong SAR, China. xmeiab@connect.ust.hk

Ming Liu is with Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China, and also with the HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen 518055, China. eelium@hkust-gz.edu.cn

Yongkang Song is with Lotus Technology Ltd, China. yongkang.song@lotuscars.com.cn

algorithms developed under this fixed input length setting cannot handle real-world applications where only partial observations are available. For instance, if a previously obscured vehicle suddenly appears in the view of the ego vehicle, the predictor's inability to accurately predict its future trajectory due to insufficient observation length can lead to a collision risk, as shown in Fig. 1. In later sections, we analyze upstream tracking and sensing data to shed light on this critical issue. We argue that even the most advanced architectures specialized for this task fail to process variable observation lengths. Alessio *et al.* [12] propose a distillation framework that recovers a reliable proxy of the same information obtained with more input observations, but it still only supports a fixed observation length. To address this problem, we propose a new hierarchical prediction framework that can handle dynamic observation lengths by introducing an SSL mask strategy to a history reconstruction pretext task. Additionally, we employ a feature distillation scheme to transfer future extraction ability to the student model. Our contributions are summarized as follows:

- Our study uncovers the critical challenge of performance degradation of trajectory predictors in the case of insufficient observations. To the best of our knowledge, this is the first comprehensive and systematic analysis of the partially observed prediction problem.
- We propose a flexible prediction framework, Partial Observations Prediction (POP), which employs Self-Supervised Learning (SSL) and feature distillation techniques, and which is capable of outputting stable and high-precision prediction results even when only partial observations are available.
- The proposed method is thoroughly evaluated on real-world datasets and a closed-loop simulator. Evaluation results demonstrate that the POP framework achieves comparable or superior performance in terms of prediction accuracy and safety metrics compared to existing state-of-the-art methods.

## II. Related Work

### A. Feature distillation in Motion Forecasting

The idea of knowledge distillation is first brought up by Hinton etal. [13] to transfer knowledge from a large, complex model (teacher) to a smaller, simpler model (student) for model compression purpose. In the context of trajectory forecasting, knowledge distillation has been used to make a model immune to incorrect detection, tracking, fragmentation, and corruption of trajectory data in crowded scenes, by distilling knowledge from a teacher with longer observation to a student with much fewer observations [12]. Although the performance is nearly retained though. This is not reasonable in more challenging autonomous driving scenes when we have longer observations(e.g. 49 frames) and only pick the first 2 frames and throw the others useful information. Inspired by [14], our aim is not to compress a model yet to improve its performance. This procedure is usually referred to as self-distillation, since the student network shares the same architecture of its teacher. Similarly to [15], our approach sets up asymmetric networks: the student is encouraged to overcome its knowledge gap by following the guide of its teacher, eventually boosting its performance. We demonstrate that knowledge distillation can lead to effective predictions even when the model has access to very few observations.

### B. SSL:Self-supervised Learning in Motion Forecasting

Self-supervised learning (SSL) has been widely explored and utilized in various research domains [16], [17]. SSL leverages the inherent structure or patterns present in unlabeled data to learn useful representations or tasks. In previous trajectory prediction work, SSL has shown promising capabilities in improving the prediction accuracy. PreM [9] focuses on connecting trajectory and scene context, enhancing their representations for trajectory forecasting. They do not have the task of reconstructing history. A recent work F-MAE [18] proposes to mask agents' trajectories and lane segments and reconstruct masked elements using a prediction head, at last fine tune the motion forecasting task. However, its reconstruction mechanism is in terms of the whole trajectory. In contrast, in our proposed method, we take the state at each time step as the reconstruction unit, which is more consistent with the real-life POP situation. A very recent work [19] used a temporal decay module to estimate the missing observation, and treat the imputation as a training objective that is jointly optimized with the motion forecasting task. However, it neglects the effectiveness of reconstructing missing observations to planning task. This is extremely vital for safe autonomous driving systems, and we will elaborate on this later in the experimental module.

## III. Problem Formulation

We adopt a structured vectorized representation to depict the map and agents. We denote the past trajectories of agents as $X_H = \{x_i\}$, where $x_i \in \mathbb{R}^{T_H \times D}$ indicating the location, yaw angle, and velocity of agent $i$ at previous $T_H$ time steps. The road map is denoted as $L = \{l_i\}$, where $l_i \in \mathbb{R}^{N \times F}$ representing $i_{th}$ lane has $N$ segments and each segment has $F$ lane semantic attributes (e.g., intersections and crosswalks). The forecasting task aims to generate $T_F$ steps future trajectories :

$$Y_F = f(X_H * M_H, L), \tag{1}$$

where $M_H = \{m_i\}$, $m_i \in \mathbb{R}^{T_H \times 1}$ indicating the validity for history state at previous $T_H$ time steps. In contrast to the previous definition of trajectory prediction, we consider that the history trajectory of the focal agent does not necessarily satisfy completeness, and thus we set $m_i^{T_H} = 0$ for states that are not valid at $T_H$.

## IV. Methodology

### A. Overview

The overview framework is shown in Fig. 2. Our prediction framework comprises three stages. In the first stage,

we train a teacher model using complete observations. The inference stage is the standard prediction process: $h_a$ and $h_m$ features are generated using road maps and history states through an encoder consisting of a multi-layer perceptron (MLP) and a location embedding layer. A series of attention modules are used to capture the interaction information between elements, and the decoders generate the initial future guess and refinement. During the SSL stage, we enable a mask procedure and use partial observations as input, along with a history reconstruction pre-task to reconstruct missing observations. In the Distillation step, we freeze the teacher model's parameters and the feature distillation strategy is used to transfer knowledge to intermediate features of partial observations, while keeping the mask procedure on. It is worth noting that, unlike previous approaches adopted SSL that aim to improve predictor performance with complete observations, we consider the distillation task with partial observations as the final fine-tuning task.

### B. Motion Forecasting Stage

In this stage We follow the inference pipeline of QCNet [4] and consider it as a strong baseline in experiments in the later section. We build a local spacetime coordinate system for each scene element to encode scene representations. These representations are transformed to Fourier features and passed through an MLP to obtain relative positional embeddings $R$. Factorized attention and self-attention mechanisms are applied to $\{X_H, R\}$ and $\{L, R\}$ respectively to obtain hidden encoded features $ha$ and $hm$. We adopt the architecture of detection transformer in both the initial and refinement decoder modules to address the one-to-many problem, allowing multiple learnable queries to cross-attend the scene encodings and a MLP to decode trajectories. A slight difference in refinement decoder is that a gated recurrent unit is used to embed each trajectory anchor, and we take its final hidden state as the mode query. Taking the output of the proposal module as anchors, we let the refinement module predict the offset to the proposed trajectories and estimate the likelihood of each hypothesis.

### C. Self-supervised learning Stage

Recall the motivation, our aim is to make predictors robust to insufficient observations. An intuitive to achieve this goal is to add noise or perform data augmentation. Since the partial observation phenomenon varies from time to time, we are not accessible to a uniform real-world noise distribution. Thus we build a pretext task from reconstructing the random masked input history. Specifically, a mask procedure and a reconstruction branch is built when performing the SSL stage. It is similar to the future prediction branch structurally. It only differs from the output dimension of decoder head, which means we use $T_H$ as the prediction horizon.

### D. Feature Distillation Stage

In order to maintain the predictive capabilities of teachers in the face of insufficient observations, our training strategy involves transferring knowledge from the entire input sequence. To achieve this, we manipulate hidden features from the encoder and interaction module. Specifically, we ensure that all of the student's features correspond to those in the teacher network. This is accomplished through a feature distillation loss, which is defined as the mean squared error (MSE) between the two feature representations:

$$\mathcal{L}_{\mathcal{D}} = \frac{1}{d} \sum \left\| (h^T - h^S) \right\|^2, \tag{2}$$

where $d$ is the dimension of the corresponding feature representations, $h^S = \{h_a^S, h_m^S, h_f^S\}$, $h^T = \{h_a^T, h_m^T, h_f^T\}$. The loss encourages the student model to learn feature representations that are similar to those of the teacher model.

### E. Training Objectives

Our goal is to build, in three steps, a model capable of accurately predicting future locations when only partial observations are available. To train the teacher model in the first stage, we employ negative log-likelihood loss and winner-take-all training strategy to optimize the best-predicted future trajectory. The training loss in this stage is:

$$\mathcal{L}_{\mathcal{MF}} = \mathcal{L}_{init} + \mathcal{L}_{refine} + \alpha \mathcal{L}_{cls}, \tag{3}$$

where the classification loss is added to optimize the mixing coefficients and $\alpha$ is parameter to balance regression and classification.

$$\mathcal{L}_{\mathcal{SSL}} = \mathcal{L}_{\mathcal{MF}} + \beta \mathcal{L}_{recons}, \tag{4}$$

$$\mathcal{L}_{\mathcal{FD}} = \mathcal{L}_{\mathcal{MF}} + \lambda \mathcal{L}_{D}. \tag{5}$$

When perform SSL stage or feature distillation stage, we keep the $L_{MF}$ and add a reconstruction loss or distillation loss with balance parameter $\beta$ or $\lambda$ respectively as shown in Eq. 4 and Eq. 5.

## V. Preliminary Analysis

Before diving into the experiment section, we will explore two questions to recall the motivation and further validate the effectiveness of the proposed method. *First of all, is it common to see inefficient observation situations in the real world? Secondly, are existing state-of-the-art (SOTA) trajectory prediction methods able to handle the partial observation problem?*

### A. Observation Distribution Analysis

To answer the first question, we investigate the Av1 tracking dataset, which is a collection of 113 log segments with 3D object tracking annotations. These log segments vary in length from 15 to 30 seconds and collectively contain a total of 11,052 tracks. We simulate the original observations using the tracking baseline method [20], which won first place on the Argoverse 3D tracking text set. According to the Av1 motion forecasting challenge standard, 20 frames are used for a complete observation. The observations are constructed frame by frame from the beginning of the appearance of the target until the target disappears. Fig. 3 shows the distribution of the observation, indicating that the prediction algorithm for autonomous driving applications is

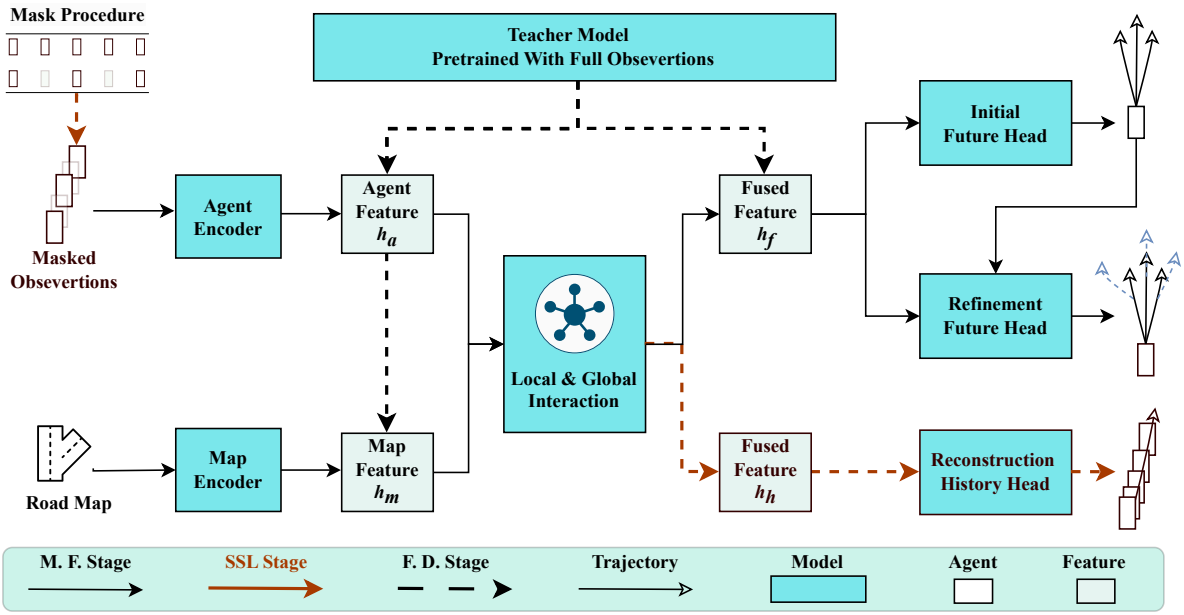


Fig. 2. **Overview of POP.** Our method consists of three stages. The motion forecasting stage involves training a teacher model with complete observations. The SSL stage consists a mask procedure and a history reconstruction pre-task. During the distillation stage, the teacher model's parameters are frozen, and a feature distillation strategy is applied to the hidden features.

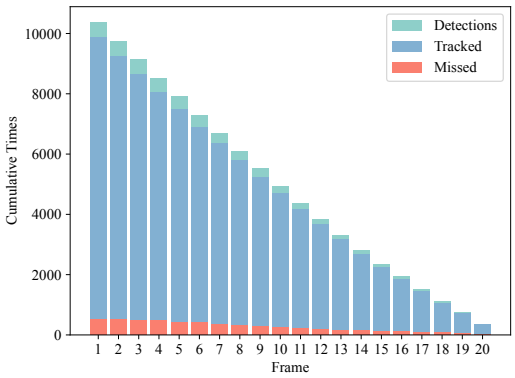


Fig. 3. **Distribution of observations with standard 20 frames.** The gray bars represent the detection of each frame for a fixed observation period, while the blue bars represent tracking. The Red indicates tracking failures.

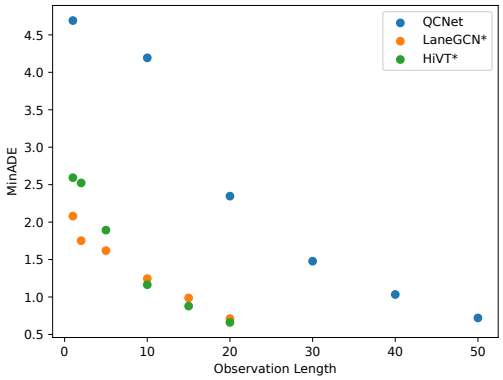


Fig. 4. **Observation length evaluation analysis on Av1 and Av2.** Methods that are evaluated on Av2 are marked with symbol "*".

often unable to satisfy the complete 20-frame observations on the dataset due to occlusion, limitations in sensing range, large speed differences between vehicles, *etc*.

### B. Observation Length Evaluation Analysis

We evaluate the $\text{MinADE}_5$ performance of three popular predictors by given observations of varying lengths. As shown in Fig. 4, the predictor's performance is directly correlated with the length of the observations. In other words, the more information from observations used as input, the greater the accuracy of the predictions made by the predictor. This relationship is especially pronounced when dealing with long prediction horizon task. Therefore, we conclude that the currently available SOTA prediction methods are unable to effectively handle situations with partial observations.

## VI. Experimental Results

### A. Open-Loop Experiment

*1) Dataset:* We evaluate our method on both Av1 and Av2. The former requires a 2s history input and a 3s future trajectory output, and the latter one focuses on a long term 6s prediction and a longer corresponding 5s history input.

*2) Metrics:* For the open-loop evaluation, we use the official benchmark metrics, including $\text{MinADE}_k$, $\text{MinFDE}_k$, $\text{MR}_k$, and brier-$\text{MinFDE}_k$. These metrics are calculated based on the trajectory with the closest endpoint to the ground truth over $k$ predictions.

*3) Implementation Details:* For all experiments, we utilize cosine learning rate decay with a weight decay of 0.0001 and an initial learning rate of 0.0005. The model is trained on 4 RTX3090s, and the pre-trained teacher model is obtained from the officially released checkpoint [4]. The hyperparameters $\alpha$, $\beta$, $\lambda$ are set to 1, 0.5, 0.5 respectively. As road

structures and traffic rules vary across regions, a generalized observation distribution cannot be obtained. We adopt a random drop scheme during training. No data augmentation or model ensemble techniques are employed.

*4) Comparison with State-of-the-Art:* We compare our method to the SOTA without ensembles on the Av2 test set in the single-agent setting, and the results are reported in Table III (upper group). Our method achieves results comparable to the best-performing methods, ranking third on the $\text{MinADE}_6$ metric. Our performance is also comparable to that of F-MAE, which uses the SSL strategy with complete observations. In later experiments, we will demonstrate the superiority of POP under partial observation conditions. To evaluate the generalization of our method, we conduct experiments on the Av1 test set using HiVT as the backbone. Although our POP-H performance is slightly degraded compared to HiVT, this is due to the use of incomplete observations to train the predictor, which increases the difficulty of fitting the network. Nonetheless, our performance is still comparable to SOTA methods, such as LTP and mmTrans.

TABLE I
CLOSED-LOOP SIMULATION RESULTS ON COMMONROAD DATASET.

| Method | P-K | DIST↑ | JERK↓ | RC↓ | CT↓ | RCT↓ |
|---|---|---|---|---|---|---|
| HiVT | 1 | 30.19 | 3.23 | 0.657 | 7 | 1 |
| HiVT | 3 | 28.80 | **2.87** | 0.636 | 2 | 1 |
| POP-H | 1 | **30.97** | 4.07 | 0.641 | 5 | **0** |
| POP-H | 3 | 28.38 | 3.07 | **0.616** | **1** | 1 |
| HiVT | 1 | 40.77 | **3.24** | 0.608 | 21 | 5 |
| POP-H | 1 | **41.35** | 3.50 | **0.576** | **17** | **4** |

The top group records performance in 91 highly interactive scenarios, while the bottom group represents that in all.

TABLE II
OPEN-LOOP SIMULATION RESULTS ON COMMONROAD DATASET.

| Model | $\text{MinADE}_1$ | $\text{MinFDE}_1$ | $\text{RMinADE}_1$ | $\text{RMinFDE}_1$ |
|---|---|---|---|---|
| HiVT | **2.51** | **5.21** | 3.22 | 8.18 |
| POP-H | 2.54 | 6.23 | **2.66** | **6.30** |

$\text{RMinADE}_1$, $\text{RMinFDE}_1$ are collected with randomly observation lengths.

*5) Ablation Study:* The ablation study results are presented in Table IV. It shows that QCNet performs poorly with partial observations. In particular, when fewer than 30 frames are available, the prediction error increases to two times compared to when the full observations are input. However, when using either the SLL or Distillation strategies individually, the performance shows a notable improvement in the POP case. This highlights the effectiveness of both stages in our design. While the SLL-only strategy achieves the same level of performance as the POP strategy with complete observations, POP outperforms SLL when there are less than 50 observations. Our findings suggest that feature distillation, treated as a fine-tuning task after SLL, allows for further feature learning and better performance.

### B. Closed-Loop Experiment

*1) Simulation Setting:* The closed-loop experiments are conducted in the interactive scenarios from CommonRoad [21], with a simulation setup similar to [22]. Considering the limited field-of-view and perception range of the autonomous vehicle (AV) in real-world scenarios, we limit our predictions to neighboring vehicles within a 50m radius. At each step, we use POP-H or HiVT to predict the future trajectories of agents over a 6s horizon with a 0.5s interval. The planning process for collision check directly incorporates the $K$ most probable prediction outcomes for each agent. The prediction model is pre-trained using feature data extracted from the closed-loop simulation itself. Each simulated scenario incorporates a designated task route to evaluate the planning performance, as shown in Fig. 5. The collision avoidance planner guides the AV along the provided task route. If the implemented algorithm fails to identify a valid solution, the AV executes a stop behavior along the generated path with a deceleration of $-4.0\,m/s^2$. All other agents are controlled by the intelligent driver model.

*2) Metrics:* We adopt the following metrics: DIST: Average completion distance of the AV along the given route in each scenario. JERK: Average jerk cost reflecting the planned trajectory's smoothness. RC: Reaction cost of other traffic agents, defined as the average deceleration efforts of nearby agents within a 40-meter range. CT: Total number of valid collision times experienced by the AV, excluding collisions at the rear and collisions with agents when the AV is stationary. RCT: Collision times at the rear of the AV.

*3) Quantitive Results:* We present the open-loop prediction performance of HiVT and POP-H in Table II, focusing on $\text{MinADE}_1$ and $\text{MinFDE}_1$ since we use the most likely prediction trajectory for collision checking in the closed-loop simulation. Results show that vanilla HiVT's performance significantly drops with random observations, while POP-H remains stable under both complete and incomplete observations due to our design for handling incomplete observations during training. The closed-loop results are reported in Table I. For each metric, the best result is in bold. As shown in the bottom group, our proposed method outperforms HiVT in almost all metrics, particularly in safety, with a 25% reduction in collisions. HiVT yields slightly better jerk results but at the cost of less driving distance and a higher number of collisions. POP-H achieves favorable performance in 91 high interactivity scenarios, particularly when using 3 predicted trajectories for collision detection. What's more, POP-H reduces collisions to 1 and achieves the lowest RC, facilitating friendly driving to other vehicles.

*4) Qualitative Results:* We demonstrate a simulation scenario where the AV navigates through a congested traffic intersection, as shown in Figure 5. During the initial phase of the simulation, the AV intends to traverse the intersection with a planned speed of 6.3 m/s. However, due to insufficient observation, the HiVT predictor inaccurately predicts the future trajectory in the first two frames. As a result, the AV fails to account for the movement of the vehicle below and begins to accelerate. By frame 3, the speed has already reached 8.7 m/s, making it too late to decelerate and leading to a collision. In contrast, POP-H consistently provides more reasonable predictions (indicated by the black scatter line) from frame 1 to frame 5, ensuring a higher level of safety.

TABLE III

COMPARISON WITH STATE-OF-THE-ART METHODS ON ARGOVERSE TEST SET.

| Model | $\text{MinADE}_6$ | $\text{MinADE}_1$ | $\text{MinFDE}_6$ | $\text{MinFDE}_1$ | brier-$\text{MinFDE}_6$ | $\text{MR}_1$ | $\text{MR}_6$ |
|---|---|---|---|---|---|---|---|
| GANet [23] | 0.72 | 1.77 | 1.34 | 4.48 | 1.96 | 0.17 | 0.59 |
| MTR [24] | 0.73 | 1.74 | 1.44 | 4.39 | 1.98 | 0.15 | 0.58 |
| GoRela [25] | 0.76 | 1.82 | 1.48 | 4.62 | 2.01 | 0.22 | 0.66 |
| F-MAE [18] | 0.71 | 1.74 | 1.39 | 4.36 | 2.03 | 0.17 | 0.61 |
| QCNet [4] | 0.65 | 1.69 | 1.29 | 4.30 | 1.91 | 0.16 | 0.59 |
| POP-Q (ours) | 0.72 | 1.86 | 1.46 | 4.84 | 2.08 | 0.20 | 0.61 |
| LaneGCN [7] | 0.87 | 1.71 | 1.36 | 3.78 | 2.05 | 0.59 | 0.16 |
| mmTrans [26] | 0.84 | 1.77 | 1.34 | 4.00 | 2.03 | 0.61 | 0.15 |
| LTP [27] | 0.83 | 1.62 | 1.30 | 3.55 | 1.86 | 0.56 | 0.15 |
| HiVT [5] | 0.77 | 1.60 | 1.17 | 3.53 | 1.84 | 0.55 | 0.13 |
| ADAPT [28] | 0.79 | 1.59 | 1.17 | 3.50 | 1.80 | 0.54 | 0.12 |
| POP-H (ours) | 0.83 | 1.73 | 1.32 | 3.83 | 1.99 | 0.59 | 0.15 |

TABLE IV

ABLATION STUDY OF MINADE/MINFDE AMONG DIFFERENT TRAINING STRATEGIES ON ARGOVERSE VALIDATION SET.

| Training strategy | | | $\text{MinADE}_6$/$\text{MinFDE}_6$ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Scratch | SSL | Distill. | Obs. = 1 | Obs. = 10 | Obs. = 20 | Obs. = 30 | Obs. = 40 | Obs. = 50 | Obs. = random |
| ✓ | | | 4.69/8.79 | 4.28/7.52 | 2.48/4.78 | 1.54/3.13 | 1.05/2.08 | **0.72/1.25** | 2.37/4.43 |
| ✓ | ✓ | | 4.03/7.45 | 2.31/4.27 | **1.10/2.05** | 0.93/1.72 | 0.85/1.57 | 0.79/1.45 | 1.47/2.72 |
| ✓ | | ✓ | 4.04/7.48 | 2.37/4.37 | 1.19/2.25 | 1.00/1.88 | 0.93/1.75 | 0.88/1.66 | 1.52/2.83 |
| ✓ | ✓ | ✓ | **4.03/7.44** | **2.24/4.16** | 1.12/2.08 | **0.92/1.70** | **0.84/1.54** | 0.79/1.45 | **1.42/2.61** |

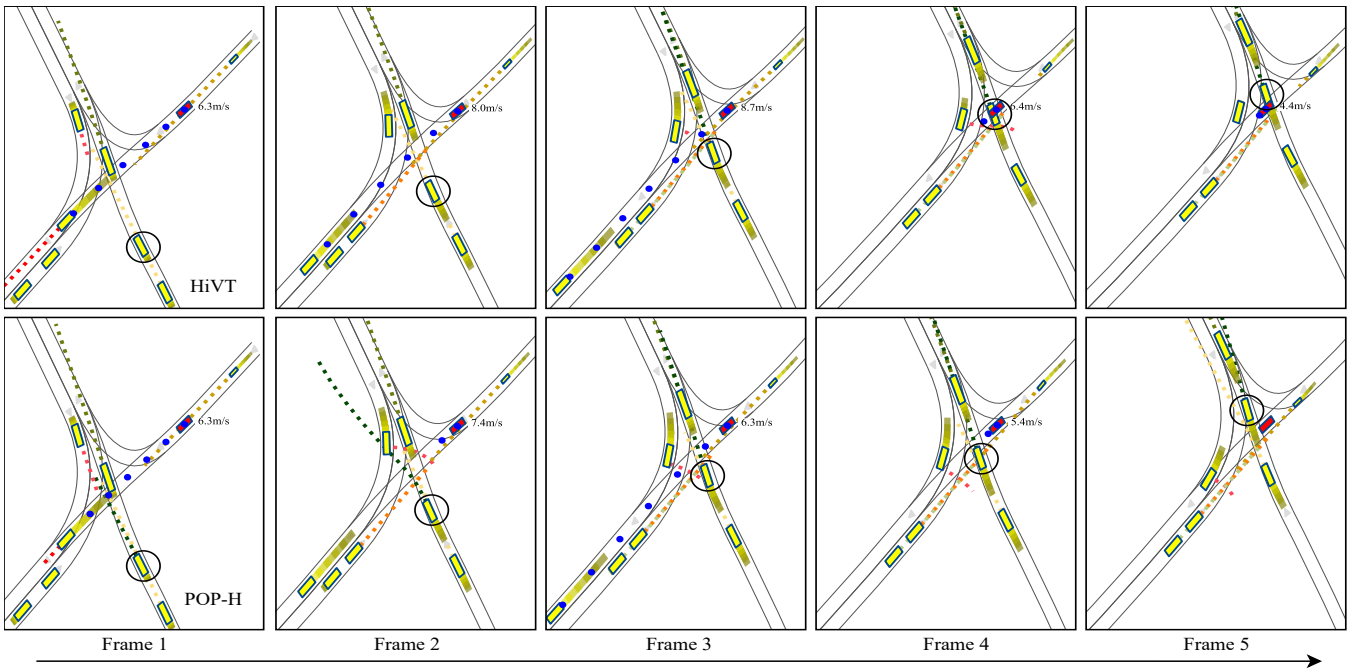


Fig. 5. **A collision scenario caused by limited observations.** The AV (red) is surrounded by yellow vehicles, the blue scatter line is the planned trajectory of the AV, and the predictions of the AV for other vehicles are marked with scatter lines of other colors. Due to limited observations, the HiVT predictor inaccurately predicted the future trajectory of a vehicle (black circle), causing the AV to accelerate and ultimately collide. In contrast, the AV equipped with the POP-H predictor exhibited superior predictions, ensuring safety.

## VII. CONCLUSIONS

We present a novel prediction framework called POP, which first employs self-supervised pre-training to help the model learn to reconstruct history, and then utilizes feature distillation as a fine-tuning task to transfer knowledge from the teacher model, which has been pre-trained with complete observations, to the student model, which has only few observations. The experiments show that compared with the existing state-of-the-art predictors, our method is able to achieve high and stable open-loop prediction accuracy both in the case of complete observations and few observations. Moreover, our method significantly enhances the safety of the autonomous driving system in the closed-loop simulation. One possible future effort is to consider exploring which observations would introduce serious hazards.

## REFERENCES


[1] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[3] Z. Huang, H. Liu, and C. Lv, "Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 3903–3913.

[4] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[5] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning*. PMLR, 2021, pp. 895–904.

[7] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 541–556.

[8] P. Bhattacharyya, C. Huang, and K. Czarnecki, "Ssl-lanes: Self-supervised learning for motion forecasting in autonomous driving," in *Conference on Robot Learning*. PMLR, 2023, pp. 1793–1805.

[9] C. Xu, T. Li, C. Tang, L. Sun, K. Keutzer, M. Tomizuka, A. Fathi, and W. Zhan, "Pretram: Self-supervised pre-training via connecting trajectory and map," in *European Conference on Computer Vision*. Springer, 2022, pp. 34–50.

[10] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 159–15 168.

[11] Y. Cao, D. Xu, X. Weng, Z. Mao, A. Anandkumar, C. Xiao, and M. Pavone, "Robust trajectory prediction against adversarial attacks," in *Conference on Robot Learning*. PMLR, 2023, pp. 128–137.

[12] A. Monti, A. Porrello, S. Calderara, P. Coscia, L. Ballan, and R. Cucchiara, "How many observations are enough? knowledge distillation for trajectory forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6553–6562.

[13] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:7200347

[14] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3713–3722.

[15] A. Porrello, L. Bergamini, and S. Calderara, "Robust re-identification by multiple views knowledge distillation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 93–110.

[16] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?" in *international conference on machine learning*. PMLR, 2020, pp. 10 871–10 880.

[17] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and P. Yu, "Graph self-supervised learning: A survey," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2022. [Online]. Available: https://doi.org/10.1109%2Ftkde.2022.3172903

[18] J. Cheng, X. Mei, and M. Liu, "Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[19] Y. Xu, A. Bazarjani, H.-g. Chi, C. Choi, and Y. Fu, "Uncovering the missing pattern: Unified framework towards trajectory imputation and prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9632–9643.

[20] J. Lambert, "Open argoverse cbgs-kf tracker," https://github.com/johnwlambert/argoverse_cbgs_kf_tracker, 2020.

[21] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726.

[22] Y. Chen, J. Cheng, L. Gan, S. Wang, H. Liu, X. Mei, and M. Liu, "Ir-stp: Enhancing autonomous driving with interaction reasoning in spatio-temporal planning," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2024.

[23] M. Wang, X. Zhu, C. Yu, W. Li, Y. Ma, R. Jin, X. Ren, D. Ren, M. Wang, and W. Yang, "Ganet: Goal area network for motion forecasting," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1609–1615.

[24] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.

[25] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun, "Gorela: Go relative for viewpoint-invariant motion forecasting," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7801–7807.

[26] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7577–7586.

[27] J. Wang, T. Ye, Z. Gu, and J. Chen, "Ltp: Lane-based trajectory prediction for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 134–17 142.

[28] G. Aydemir, A. K. Akan, and F. Güney, "ADAPT: Efficient multi-agent trajectory prediction with adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.