IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

MacFormer: Map-Agent Coupled Transformer for Real-time and Robust Trajectory Prediction

Chen Feng[2], Hangning Zhou[3], Huadong Lin[3], Zhigang Zhang[3], Ziyao Xu[3], Chi Zhang[3], Boyu Zhou[1,†], and Shaojie Shen[2]

***Abstract*—Predicting the future behavior of agents is a fundamental task in autonomous vehicle domains. Accurate prediction relies on comprehending the surrounding map, which significantly regularizes agent behaviors. However, existing methods have limitations in exploiting the map and exhibit a strong dependence on historical trajectories, which yield unsatisfactory prediction performance and robustness. Additionally, their heavy network architectures impede real-time applications. To tackle these problems, we propose Map-Agent Coupled Transformer (MacFormer) for real-time and robust trajectory prediction. Our framework explicitly incorporates map constraints into the network via two carefully designed modules named coupled map and reference extractor. A novel multi-task optimization strategy (MTOS) is presented to enhance learning of topology and rule constraints. We also devise bilateral query scheme in context fusion for a more efficient and lightweight network. We evaluated our approach on Argoverse 1, Argoverse 2, and nuScenes real-world benchmarks, where it all achieved state-of-the-art performance with the lowest inference latency and smallest model size. Experiments also demonstrate that our framework is resilient to imperfect tracklet inputs. Furthermore, we show that by combining with our proposed strategies, classical models outperform their baselines, further validating the versatility of our framework.**

***Index terms*— Deep Learning Methods; Representation Learning; Autonomous Vehicle Navigation**

I. Introduction

ACCURATE trajectory prediction of nearby agents is crucial for safe navigation of autonomous vehicles. Understanding the surrounding map is essential to enhance prediction performance, as it imposes topology and rule constraints greatly regularizing agent behaviors. Specifically, the topology constraint requires consistency with lane types, while the rule constraint prohibits driving outside drivable areas and prefers trajectories near centerlines.

To integrate map constraints into models, existing works utilize the map mainly in two manners: *implicit-fusion* and *target-driven*. The *implicit-fusion* models [1]–[10] use implicit frameworks to fuse the historical motions of maps and agents into high-level latent features, which are then decoded to predict multiple trajectories. However, to achieve high-level latent features, a deeper network design is necessary, resulting in a heavier architecture. Specifically, multiple layers are typically stacked in the context fusion stage to effectively fuse map information and agent motions. Meanwhile, during training, these models only minimize errors between ground-truth and predicted trajectories without explicitly using map constraints. This results in the model output being dominated by past motions and ignoring map information due to high correlation between historical and future movements [11]. Thus, these models are likely to produce inadequate probabilistic predictions or infeasible trajectories, *e.g.* causing mode collapse or driving outside the drivable area.

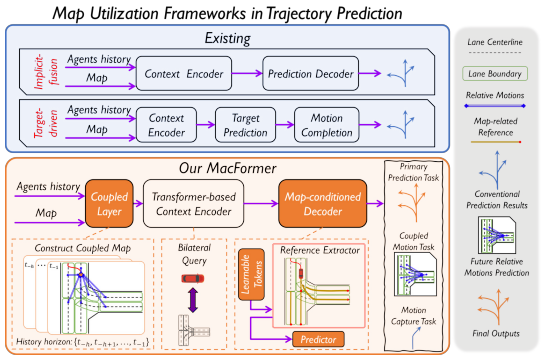

Fig. 1. **The overview of map utilization in trajectory prediction.** Existing works (Top) take *implicit-fusion* or *target-driven* manner, which are yet to exhaust the map. In contrast, our method (Bottom) explicitly and sufficiently leverage map constraints, effectively capturing prediction uncertainty and enhancing robustness to imperfect tracklets. (Sect.I)

On the other hand, *target-driven* models [12]–[17] reduce prediction uncertainty by selecting plausible endpoint targets from the map and using them to complete future motions. Nevertheless, these models only utilize map information when choosing endpoints, and capturing future behaviors solely based on a target remains challenging. Hence, there is still significant prediction uncertainty. Additionally, these models suffer from a performance-cost trade-off, *i.e.*, fewer target candidates lead to performance decline while more candidates result in extensive computation cost. In a nutshell, current methods lack proper use of maps and rely heavily on historical trajectories, resulting in unsatisfactory prediction performance when faced with imperfect historical motions. Moreover, they employ heavy network architectures, particularly in context fusion, which may hinder real-time capability in practical applications.

To address the above-mentioned issues, we propose **Mac-**

Manuscript received: April 24, 2023; Revised: July 24, 2023; Accepted: August 7, 2023. This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by The Research Grants Council General Research Fund (RGC GRF) project RMGS20EG20.

[1]School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China.
[2]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.
[3]Megvii Research, Beijing, China.
Email: cfengag@ust.hk, zhouby23@mail.sysu.edu.cn
† **Corresponding Author**
Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

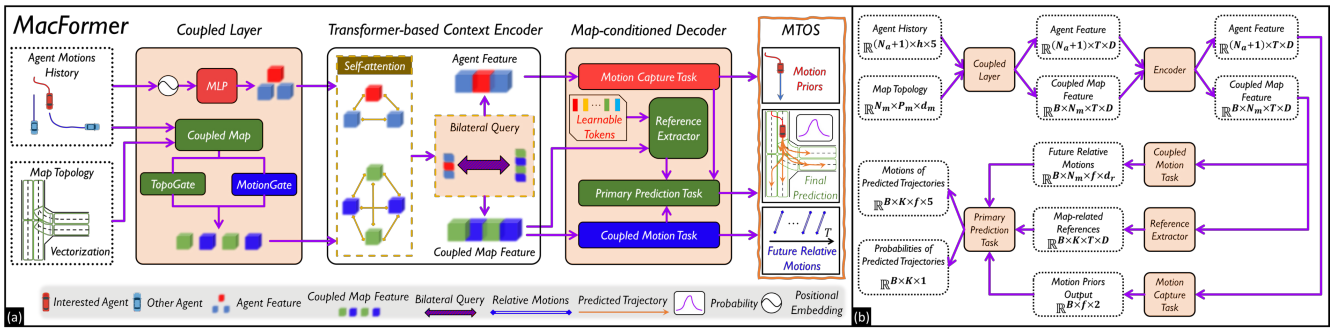

Fig. 2. (a) The system overview of **MacFormer**. (b) The detailed implementation of **MacFormer** and output size of each operation.

Former, an one-stage **M**ap-**A**gent **C**oupled Trans**Former**, which directly couples map constraints with agent motions for real-time and robust trajectory prediction (Fig.1). In order to enable direct utilization of the coupled relation between the map and agent, two modules named **coupled map** and **reference extractor** are carefully designed, which explicitly integrate map constraints into the system. Furthermore, to assure that the map constraints are well learned during training, we propose a multi-task optimization strategy (**MTOS**). It ensures map-constrained prediction by forecasting future relative motions in coupled map, capturing motion priors, and outputting final predictions with corresponding probabilities via map-conditioned regression. Lastly, to enhance computational efficiency and promote real-time prediction, a concise **bilateral query** scheme for context fusion is devised. It allows parallel cross-domain information interaction between the map and agent from their distinct perspectives and shared usage of their affinity matrix, significantly reducing the time and space complexity of context fusion.

The proposed method was evaluated on three large-scale real-world benchmarks (Argoverse 1&2 [18], [19] and nuScenes [20]) and achieved state-of-the-art performance with significantly lower inference latency and fewer parameters. We also conducted extensive experiments to validate the effectiveness and robustness of our framework. Furthermore, we introduced our framework to classical prediction models, enhancing their performance while reducing parameters. In summary, the contributions of this paper are:

1) A map-agent coupled framework for real-time and robust trajectory prediction, which efficiently and effectively integrates map constraints via coupled map, reference extractor, and multi-task optimization strategy (MTOS).

2) An efficient and lightweight context fusion scheme, bilateral query, which allows parallel context fusion between map and agent from their individual views.

3) Extensive evaluation in multiple real-world benchmarks. It shows that **MacFormer** achieves state-of-the-art performance, while it is faster and more lightweight than existing models. Experiments also demonstrate the versatility and robustness of the proposed framework, which enhances classical prediction methods while maintaining satisfactory resilience to imperfect upstream results.

II. Related Work

Map utilization. Map utilization has received much attention recently, where existing approaches mainly fall into two different directions: *implicit-fusion* and *target-driven*.

1) ***Implicit-fusion.*** *Implicit-fusion* models use different backbones to encode map and fuse it with agent motions in high-level latent feature. By optimizing prediction outputs, they aim to capture the map structure in such fusion. Early works [2], [21] used CNNs to encode the map and agents together into an image, then predicted trajectories using a fully connected layer. However, this approach cannot model relations between different scenario elements effectively. VectorNet [1], LaneGCN [3], and DSP [8] all vectorized the map and adopted graph-based architectures to extract features and interactions of agents and map instead. Besides, TPCN [7] migrated point cloud learning backbone to implement the feature fusion. To cover long-range prediction, mmTransformer [6], SceneTransformer [4] and MultiPath++ [5] leveraged the global receptive field of Transformer to capture whole context fusion. Nevertheless, *implicit-fusion* still leads to mode collapse or infeasible forecasts that violate map constraints, which illustrates their unsatisfactory map utilization and prediction stability.

2) ***Target-driven.*** To overcome the limitations in *implicit-fusion* models, TNT [12] and DenseTNT [13] decomposed prediction objectives into two stage: target prediction and motion completion. They first sampled several target candidates for the agent from map, selected qualified targets from candidates, then completed full trajectory conditioned on each target. Similarly, HOME [14], GOHOME [15] and THOMAS [16] forecasted a heatmap and greedily sampled target candidates to implement target prediction before generating full trajectories. However, despite the high intent correlation of targets, such models still cannot effectively capture motions of the entire process, while also incurring redundant costs due to large quantities of unselected candidate.

In contrast, our method tackles these problems by directly and sufficiently leveraging map, which utilizes the coupled relation by coupled map, enables efficient map-constrained predictions via reference extractor, considers map constraints during optimization via MTOS.

Transformer. Transformer [22] has been successfully applied in various fields, *e.g.*, natural language processing and computer vision. Recently, some works [4]–[6] have utilized it and its variants for trajectory prediction due to its capability of global and flexible feature extraction that are highly applicable for context interaction. To leverage this architecture, our proposed method devises a Transformer-based context encoder and proposes an effective query module in reference extractor

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

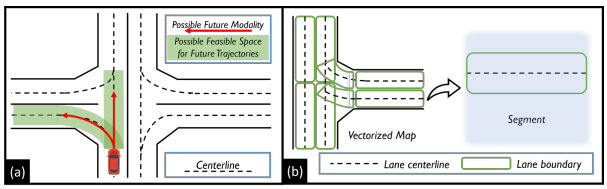

Fig. 3. (a) Illustration of map constraints on multi-modality. (b) Illustration of segments in vectorized map.

to extract map-related references from coupled map feature unlike a concurrent work MTR [10] that designs a motion query pair for learning motion modes in decoder.

III. Problem Formulation

Given the past movements of the interested agent $\mathbf{s}_H$ over h timestamps, the nearby map topology $\mathcal{M}$ (position, type, and connectivity of lanes), and observations of N_a other agents $\mathcal{S} = \{\mathbf{s}_1, ..., \mathbf{s}_{N_a}\}$, existing works aim to determine the distribution $P(\mathcal{Y}|\mathbf{s}_H, \mathcal{S}, \mathcal{M})$ for future motions represented by $\mathcal{Y} = \{y_0, ..., y_{f-1}\}$ spanning f timestamps. The entire time range is denoted as $T = h + f$ timestamps.

However, as mentioned before, the formulation of existing approaches cannot sufficiently exploit map information. Thus, we propose MTOS to effectively learn map constraints (Sect.V-C). In our formulation, the optimization objective also includes predicting future relative motions $\mathcal{R}$ and motion priors $\mathcal{J}$. Specifically, $\mathcal{M}$ is divided into several segments, each of which is discretized as a set of points. Each discrete point contains its position, lane type, and connected segments. $\mathcal{R}$ denotes the vectors pointing from the closest point in each segment towards $\mathbf{s}_H$ in the future horizon. $\mathcal{J}$ is represented as future positions of $\mathbf{s}_H$ inferred merely from historical trajectory to capture the scale and pattern of history. Hence, based on our coupled map $\mathcal{C}_{\mathcal{M}}(\mathbf{s}_H, \mathcal{M})$ that includes historical relative motions and map topology, the objective of **MacFormer** is formulated as a joint distribution $P(\mathcal{Y}, \mathcal{R}, \mathcal{J}|\mathcal{X})$, where we denote $\mathcal{X} = (\mathbf{s}_H, \mathcal{S}, \mathcal{C}_{\mathcal{M}}(\mathbf{s}_H, \mathcal{M}))$. Due to the conditional independence of $\mathcal{R}$ and $\mathcal{J}$ conditioned on $\mathcal{X}$, we factorize it as two marginal distributions and a conditional distribution to diminish the complexity of learning a joint distribution:

$$P(\mathcal{Y}, \mathcal{R}, \mathcal{J}|\mathcal{X}) = P(\mathcal{Y}|\mathcal{R}, \mathcal{J}, \mathcal{X})P(\mathcal{R}|\mathcal{X})P(\mathcal{J}|\mathcal{X}). \quad (1)$$

In fact, $P(\mathcal{Y}|\mathcal{R}, \mathcal{J}, \mathcal{X})$ is still not tractable for the model due to its high uncertainty and complicated multi-modality. As illustrated in Fig.3(a), trajectories are typically confined within a space around possible centerlines. Under this observation, the entire difficult prediction problem can be decomposed into two easier tasks: 1) predicting the possible centerlines, named map-related references $\Psi = \{\psi_1, ...\psi_K\}$ where ψ denotes one possible map-related reference. and 2) predicting trajectories conditioned on each of the map-related references. Hence, we decompose $P(\mathcal{Y}|\mathcal{R}, \mathcal{J}, \mathcal{X})$ according to law of total probability to alleviate the high uncertainty. We have $P(\psi|\mathcal{R}, \mathcal{J}, \mathcal{X})$=$P(\psi|\mathcal{X})$ due to the conditional irrelevance of ψ conditioned on $\mathcal{R}$ and $\mathcal{J}$. As demonstrated in eq.2, we firstly extract each plausible map-related reference according to agent history and map topology, represented as $P(\psi|\mathcal{X})$ (Sect.V-B). Then, each future trajectory is predicted given the corresponding reference as the guidance, *i.e.*, $P(\mathcal{Y}|\psi, \mathcal{R}, \mathcal{J}, \mathcal{X})$ (Sect.V-C). Therefore, we expect all possible ψ can capture plausible and complicated multi-modality to better facilitate trajectory prediction. This decomposition aims to narrow the prediction space, enabling each prediction near its corresponding reference.

$$P(\mathcal{Y}|\mathcal{R}, \mathcal{J}, \mathcal{X}) = \sum_{\psi \in \Psi} P(\mathcal{Y}|\psi, \mathcal{R}, \mathcal{J}, \mathcal{X})P(\psi|\mathcal{R}, \mathcal{J}, \mathcal{X}),$$
$$P(\mathcal{Y}|\mathcal{R}, \mathcal{J}, \mathcal{X}) = \sum_{\psi \in \Psi} P(\mathcal{Y}|\psi, \mathcal{R}, \mathcal{J}, \mathcal{X})P(\psi|\mathcal{X}). \quad (2)$$

Furthermore, our framework can be extended to joint prediction for all agents in the scenario. The objective is predicting $\mathcal{Y}, \mathcal{R}, \mathcal{J}$ of each agent conditioned on its corresponding coupled map $\mathcal{C}_{\mathcal{M}}$ and history of all N_a+1 agents. For all agents, all parameters in our model are shared and all operations are parallel. Thus, we define the number of predicted agents as B, B=1 for single prediction while B=N_a+1 for joint prediction (practical usage for autonomous navigation system). For brevity, we set B=1 to clearly introduce our method.

IV. System Overview

The proposed framework, as shown in Fig.2, consists of three main components. Firstly, coupled layer extracts historical motion features and constructs coupled map. It calculates historical relative motions between the map and agent in each timestamp. Then, coupled map is defined as the combination of historical relative motions and map topology (Sect.V-A). Subsequently, MotionGate and TopoGate are devised to respectively extract its temporal and spatial features. Secondly, Transformer-based context encoder employs self-attention for social interaction within respective domain (map or agent), then efficiently achieves parallel context fusion between them using our bilateral query scheme (Sect.V-D). Thirdly, map-conditioned decoder learns corresponding map-related references from coupled map via reference extractor to guide predicted trajectories spatially along some centerlines or their combination (Sect.V-B). To ensure predictions are constrained by maps, we develop multi-task optimization strategy (MTOS). Coupled motion task forecasts future relative motions in coupled map directly coupling the predictions with the map and imposing map constraints on trajectories. Motion capture task conventionally predicts one future trajectory to capture motion priors by learning scale and pattern of history. Lastly, primary prediction task regresses final predictions with corresponding probabilities conditioned on map-related references and outputs from above two tasks (Sect.V-C).

V. Methodology

A. Coupled Layer

To begin, we apply the agent-centric strategy to normalize all inputs based on the coordinate system of the interested agent. Then, coupled layer processes them for extracting motion features of agents and constructing coupled map.

In our formulation, the motions of agents $\mathbf{s}$ are represented as the set of locations (x, y), azimuth angles (α), and velocity (v) $[x, y, \cos\alpha, \sin\alpha, v]$. For the future horizon, we use a mask on these timestamps then concatenate it with agents'

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3311351

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

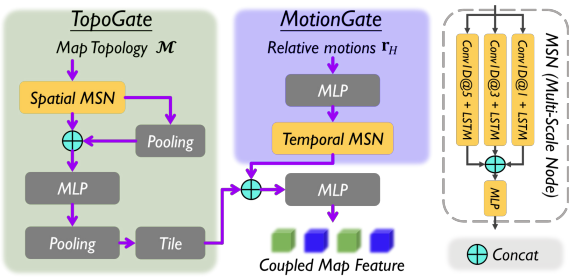

Fig. 4. The detailed structure of **TopoGate** and **MotionGate**. (Sect.V-A)

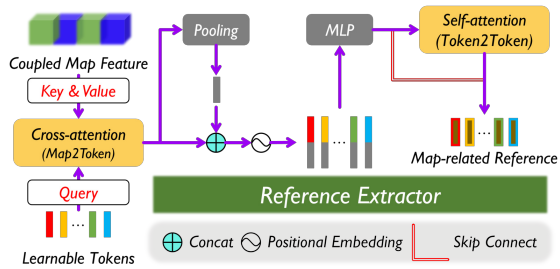

Fig. 5. The overview of the proposed **reference extractor**. (Sect.V-B)

historical motions ($\mathbf{s}_H$ and $\mathcal{S}$). Subsequently, positional embedding $\mathrm{PE}(\cdot)$ is applied on each timestamp to reinforce the identification of each timestamp [22]. Afterwards, we deploy a multi-layer perceptron network $\mathrm{MLP}(\cdot)$ for motions inputs. The agent feature in coupled layer $\mathcal{F}_A^{CL}$ follows:

$$\mathcal{F}_A^{CL} = \mathrm{MLP}(\mathrm{PE}([\mathbf{s}_H, \mathcal{S}])) \in \mathbb{R}^{(N_a+1)\times T\times D}. \tag{3}$$

with D the feature dimension and $[\cdot]$ the concatenation.

Construct Coupled Map. We have chosen the vectorized map to describe map topology $\mathcal{M}$ due to its compactness and completeness of map representation [1]. As shown in Fig.3(b), the lane area is divided into N_m segments denoted as $\mathcal{M} = \{m_i | i \in [1, ..., N_m]\}$, where each segment is discretized into a set of P_m points. Each point contains d_m attributes, *e.g.*, location, predecessor and successor points, road type, connected segments. Then, we calculate the historical relative motions $\mathbf{r}_H$ between $\mathbf{s}_H$ and $\mathcal{M}$. Here, $\mathbf{r}_H$ refers to the vectors from the closest point in each segment to the interested agent at each timestamp, including d_r attributes (distance and direction). The relative motions calculation function $\varphi(\cdot)$ follows:

$$\begin{aligned}\varphi(\mathbf{s}_H, \mathcal{M}) = \{\mathbf{s}_H - m_i(l) | l = \arg\min_p ||\mathbf{s}_H - m_i(p)||_2,\\ p \in [1, ..., P_m], i \in [1, ..., N_m]\}.\end{aligned} \tag{4}$$

where $m_i(p)$ is one discrete point of the segment m_i. Like the operations performed on agents, the relative motions $\mathbf{r}_H$ are also subject to a future horizon mask and positional embedding at each timestamp. The coupled map $\mathcal{C}_\mathcal{M}$ is then formed by combining $\mathcal{M}$ and $\mathbf{r}_H$, as illustrated below:

$$\begin{aligned}\mathbf{r}_H &= \mathrm{PE}(\varphi(\mathbf{s}_H, \mathcal{M})) \in \mathbb{R}^{B\times N_m\times T\times d_r},\\ \mathcal{C}_\mathcal{M}(\mathbf{s}_H, \mathcal{M}) &= \{\mathbf{r}_H, \mathcal{M}\}.\end{aligned} \tag{5}$$

The topology constraint requires that agent behaviors align with the lane types, namely leftmost, middle, and rightmost lanes. For instance, if agents are in the leftmost lane at an intersection, they can only turn left or proceed straight ahead because there is no lane connected to the leftmost one that allows right turns on the map topology. Thus, as depicted in Fig.4, we devise TopoGate $\mathcal{G}^T$ to effectively extract map topology where the core operation is the Multi-Scale Node (MSN). Specifically, MSN contains three 1D Conv layers with different kernel sizes followed by LSTM to facilitate multi-scale and sequential feature extraction. TopoGate adopts a spatial MSN with all operations on P_m (points) dimension for capturing spatial topology. We then tile the map topology feature by T-fold consistent with the shape $\mathbf{r}_H$ as $\mathcal{G}^T(\mathcal{M}) \in \mathbb{R}^{B\times N_m\times T\times D}$. Similarly, MotionGate $\mathcal{G}^M$ is proposed for leveraging the time sequence of historical relative motions $\mathbf{r}_H$ as shown in Fig.4. First, we deploy an MLP block for relative motion encoding. Then, temporal MSN is applied on T (time) dimension of $\mathbf{r}_H$ to extract temporal motion information resulting in $\mathcal{G}^M(\mathbf{r}_H) \in \mathbb{R}^{B\times N_m\times T\times D}$. By combining these two gates together, we obtain the coupled map feature in coupled layer $\mathcal{F}_{\mathcal{C}_\mathcal{M}}^{CL}$:

$$\mathcal{F}_{\mathcal{C}_\mathcal{M}}^{CL} = \mathrm{MLP}([\mathcal{G}^T(\mathcal{M}), \mathcal{G}^M(\mathbf{r}_H)]) \in \mathbb{R}^{B\times N_m\times T\times D}. \tag{6}$$

B. Reference Extractor

The rule constraint prohibits driving outside the drivable area, and trajectories near centerlines rather than lane boundaries are preferred. This constraint yields trajectories confined within neighborhood space surrounding the centerlines. Thus, we propose the reference extractor denoted as $\mathcal{RE}(\cdot)$ that can learn map-related references from the coupled map feature in encoder $\mathcal{F}_{\mathcal{C}_\mathcal{M}}^{Encoder}$, as illustrated in Fig.5. These references guide predictions throughout the entire future process. This module aims to model $P(\psi|\mathcal{X})$ guiding final K predicted trajectories to capture the entire prediction uncertainty and plausible multi-modality. Specifically, we generate K different learnable tokens (each one corresponds to a modality) to extract the map-related reference feature from $\mathcal{F}_{\mathcal{C}_\mathcal{M}}^{Encoder}$ by cross-attention. Notably, the information learned by tokens includes the spatial topology of map, agent motions, and their coupled relations. To ensure the consistency of multi-modality, we concatenate the pooling feature with each token. Subsequently, positional embedding is applied on each token and self-attention enhances the interaction within multi-modality to avoid mode collapse. With the aid of $\mathcal{RE}(\cdot)$, the map-related references feature $\mathcal{F}_\Psi$ capturing $P(\Psi|\mathcal{X})$ is given as:

$$\mathcal{F}_\Psi = \mathcal{RE}(\mathcal{F}_{\mathcal{C}_\mathcal{M}}^{Encoder}) \in \mathbb{R}^{B\times K\times T\times D}. \tag{7}$$

C. Multi-Task Optimization Strategy

We propose a multi-task optimization strategy (MTOS) for effectively incorporating map constraints. This strategy prevents the network from neglecting map information and enhances its ability to learn map constraints during training. We discuss each task individually in MTOS, followed by the uniform optimization process.

Coupled Motion Task. Relative motions effectively bridge the map constraints and trajectories. If the network can forecast accurate future relative motions $\mathcal{R}$, it demonstrates the network

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

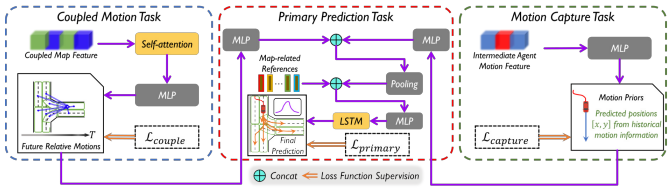

Fig. 6. Illustration of map-conditioned regression. (Sect.V-C)

has learned map constraints on agent motions. Hence, we forecast $\mathcal{R}$ from coupled map feature $\mathcal{F}_{\mathcal{CM}}^{Encoder}$:

$$\mathcal{R} = \text{MLP}(\text{SelfAttention}(\mathcal{F}_{\mathcal{CM}}^{Encoder})) \in \mathbb{R}^{B\times N_m\times f\times d_r}. \tag{8}$$

We pre-label the future states of coupled map $\mathcal{R}_{gt}$ via $\varphi(\cdot)$ defined in Sect.V-A as $\mathcal{R}_{gt} = \varphi(\mathcal{Y}_{gt}, \mathcal{M})$. $\mathcal{Y}_{gt}$ is the future trajectory ground-truth. Thus, the loss term of this task is:

$$\mathcal{L}_{couple} = ||\mathcal{R} - \mathcal{R}_{gt}||_2^2. \tag{9}$$

Motion Capture Task. The objective of this task is to allow the network to capture the motion priors $\mathcal{J}$ of agent inputs *e.g.*, scale and pattern of trajectory. As stated in Sect.III, $\mathcal{J}$ is represented as the agent's positions $[x, y]$ inferred only using feature extracted solely from agent motion $\mathcal{F}_A^{Encoder}$:

$$\mathcal{J} = \text{MLP}(\mathcal{F}_A^{Encoder}) \in \mathbb{R}^{B\times f\times 2}. \tag{10}$$

We denote the errors between $\mathcal{J}$ and positions in $\mathcal{Y}_{gt}$ as $E = \mathcal{J} - \mathcal{Y}_{gt}\{x, y\}$ and the loss function for this task:

$$\mathcal{L}_{capture} = \begin{cases} 0.5E^2 & \text{if } |E| < 1 \\ |E| - 0.5 & \text{otherwise.} \end{cases} \tag{11}$$

Primary Prediction Task. This task gives the final predictions of our framework. According to Sect.III, we devise map-conditioned regression for explicitly utilizing map constraints. Specifically, we first concatenate the features of future relative motions $\mathcal{R}$ and motion priors $\mathcal{J}$, denoted as $\mathcal{Z}$. Then, the motions of predicted trajectories $\mathcal{Y}$ is conditionally regressed based on $\mathcal{Z}$ and map-related references feature $\mathcal{F}_\Psi$:

$$\begin{aligned} \mathcal{Z} &= [\text{MLP}(\mathcal{R}), \text{MLP}(\mathcal{J})] \in \mathbb{R}^{B\times N_m\times f\times D}, \\ \mathcal{Y} &= \text{LSTM}(\text{MLP}([\mathcal{F}_\Psi, \text{Pooling}(\mathcal{Z})])). \end{aligned} \tag{12}$$

The regressed motions are $\mathcal{Y} \in \mathbb{R}^{B\times K\times f\times 5}$ with five attributes $[x, y, \cos\alpha, \sin\alpha, v]$ as the same as input. To ensure smoothness within future trajectories, they are decoded via LSTM [23] which allows prediction of each point is conditioned on its previous predicted points. Besides, we adopt maximum entropy model to predict the probabilities $P(\mathcal{Y})$ of all the K trajectories as:

$$P(\mathcal{Y}) = \frac{\exp(\text{MLP}(\mathcal{F}_\Psi))}{\sum_{k=1}^{K}\exp(\text{MLP}(\mathcal{F}_{\Psi k}))}. \tag{13}$$

with $P(\mathcal{Y}) \in \mathbb{R}^{B\times K\times 1}$ and k the indexer of tensor.

To train them with given ground-truth $\mathcal{Y}_{gt}$, we use the Gaussian Mixture Model (GMM) loss for regression and max-margin loss for probability:

$$\begin{aligned} \mathcal{L}_{gmm}(r, p, r_{gt}) &= -\log\sum_{i=1}^{K} p_i e^{-\frac{1}{2}||r^i - r_{gt}^i||_2^2}, \\ \mathcal{L}_m(p) &= \frac{1}{K-1}\sum_{i=1, i\neq\bar{i}}^{K}\max(0, p_i + \delta - p_{\bar{i}}). \end{aligned} \tag{14}$$

where r, p, r_{gt} represent motions of predicted trajectories, predicted probability, and trajectory ground-truth accordingly. The $p_{\bar{i}}$ is the probability of the prediction closest to the ground-truth. Additionally, δ is a margin set at $\frac{1}{K}$. Hence, we define the loss term for primary prediction task:

$$\mathcal{L}_{primary} = \mathcal{L}_{gmm}(\mathcal{Y}, P(\mathcal{Y}), \mathcal{Y}_{gt}) + \mathcal{L}_m(P(\mathcal{Y})). \tag{15}$$

Uniform Optimization Process. The supervision of the three tasks mentioned above results in a one-stage and fully supervised end-to-end training. The total loss function, which includes eq.9, 11, and 15, is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{primary} + \mathcal{L}_{couple} + \mathcal{L}_{capture}. \tag{16}$$

Through minimizing eq.16, MTOS achieves the uniform optimization of the objective joint distribution $P(\mathcal{Y}, \mathcal{R}, \mathcal{J}|\mathcal{X})$.

D. Bilateral Query

The proposed method differs from existing ones in that it incorporates a bilateral query strategy, which efficiently facilitates cross-domain interaction. Unlike the unilateral query that only supports each agent queries its interested region on the map, our scheme allows mutual querying of information between map and agents for more effective context fusion. Specifically, compared to existing methods, we add an operation that each segment in coupled map feature captures corresponding agent motions within their respective range. Given the agent motion feature $\mathcal{F}_A^{SI}$ and coupled map feature $\mathcal{F}_{\mathcal{CM}}^{SI}$ after social interaction, we first calculate the affinity matrix $\mathbf{M}_{aff}$ between agents and coupled map using the learnable linear projection matrix $\mathbf{W}_{bq}$ and dot-product,

$$\begin{aligned} \mathbf{z}^{agent} &= \mathbf{W}_{bq}\mathcal{F}_A^{SI}, \mathbf{z}^{map} = \mathbf{W}_{bq}\mathcal{F}_{\mathcal{CM}}^{SI}, \\ \mathbf{M}_{aff} &= \mathbf{z}^{agent}\mathbf{z}^{map} \in \mathbb{R}^{(N_a+1)\times N_m}. \end{aligned} \tag{17}$$

Then, we implement the parallel bilateral query process from their respective views based on affinity matrix by the scaled element-product,

$$\begin{aligned} \mathbf{q}^{agent} &= \mathbf{W}_Q^{agent}\mathcal{F}_A^{SI}, \mathbf{q}^{map} = \mathbf{W}_Q^{map}\mathcal{F}_{\mathcal{CM}}^{SI}, \\ \mathbf{v}^{agent} &= \mathbf{W}_V^{agent}\mathcal{F}_A^{SI}, \mathbf{v}^{map} = \mathbf{W}_V^{map}\mathcal{F}_{\mathcal{CM}}^{SI}. \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbf{h}^A &= \text{softmax}(\frac{\frac{1}{N_a+1}\sum_{i=0}^{N_a}\mathbf{q}_i^{agent}}{\sqrt{D}} \odot \mathbf{M}_{aff})\mathbf{v}^{map}, \\ \mathbf{h}^{\mathcal{CM}} &= \text{softmax}(\frac{\frac{1}{N_m}\sum_{j=0}^{N_m-1}\mathbf{q}_j^{map}}{\sqrt{D}} \odot \mathbf{M}_{aff}^{\mathsf{T}})\mathbf{v}^{agent}. \end{aligned} \tag{19}$$

where $\mathbf{W}_Q^{agent}$, $\mathbf{W}_Q^{map}$, $\mathbf{W}_V^{agent}$, $\mathbf{W}_V^{map}$ are learnable linear projection matrices matrices for query and value. The symbol $\odot$ is element-wise product while i and j denote the indexer of tensor. Afterwards, we obtain agent motion feature $\mathcal{F}_A^{Encoder}$ and coupled map feature $\mathcal{F}_{\mathcal{CM}}^{Encoder}$ after context fusion similar to Feed-Forward Network in Transformer [22]:

$$\begin{aligned} \mathcal{F}_A^{Encoder} &= \text{MLP}(\text{LayerNorm}(\mathbf{h}^A + \mathcal{F}_A^{SI})), \\ \mathcal{F}_{\mathcal{CM}}^{Encoder} &= \text{MLP}(\text{LayerNorm}(\mathbf{h}^{\mathcal{CM}} + \mathcal{F}_{\mathcal{CM}}^{SI})). \end{aligned} \tag{20}$$

The bilateral query serves the same function as a 2-layer cross-attention, but with a more compact and parallel structure. This is achieved by using a shared affinity matrix $\mathbf{M}_{aff}$, resulting in reduced time and space complexity. Additionally, our module can be adapted to multi-head form similar to the original Transformer architecture.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

VI. EXPERIMENTS

A. Experiment Setup

Real-world Datasets. We train and evaluate our model on three large-scale and challenging real-world datasets. Argoverse 1 [18] contains 333k 5-second sequences where each corresponds to a specific scenario sampled from a 290km-long roadway at 10 Hz. They provide trajectory histories, other agents and HD maps, with $(0, 2]$ seconds for observation and $(2, 5]$ seconds for prediction. Argoverse 2 [19] focuses on the prediction of multiple road users (vehicle, pedestrian, motorcyclist, cyclist and bus) with 250k 11-second scenarios sampled from 2110 km over six geographically diverse cities at 10 Hz. The dataset gives 5 seconds observation to predict the behaviors in the future 6 seconds. nuScenes [20] is collected from 1000 scenes in Boston and Singapore, containing over 40000 samples, published at 2 Hz. The forecasting task for models is to predict the next 6 seconds according to the trajectory histories and HD maps in the past 2 seconds.

Metrics. We use the extensively adopted official metrics. 1) minFDE_K: the minimum endpoint error between K trajectories and ground-truth. 2) minADE_K: the minimum average displacement error of each point between K trajectories and ground-truth. 3) MR_K: the proportion of scenarios where none of the predictions' endpoints are within 2.0 meters of ground-truth. 4) brier-score: $(1.0\text{-}p)^2$ with p as the probability of the trajectory with minimum endpoint error. On Argoverse, we also report the official ranking metric brier-minFDE_K, indicating the minFDE_K plus brier-score.

Implementation Details. We first divide the lane area within radius 50m from the interested agent into N_m=128 segments, which contains up to P_m=31 points with d_m=15 attributes. Within this area, we select N_a=31 agents closest to the interested agent. The relative motions in coupled map are defined as a d_r=3 vector $[dist, \cos\beta, \sin\beta]$ presenting distance ($dist$) and direction (β). All attention layers contain 4 heads and 128 hidden units. Notably, the quantity of learnable tokens and predicted trajectories K is set 6 on Argoverse 1 and 2 [18], [19] while 10 on nuScenes [20]. Our model is trained for 200 epochs on 8 NVIDIA RTX 2080Ti GPUs. We use Adam optimizer with the initial learning rate of 1e-4, decaying to 1e-5 at 170^{th} epoch and 1e-6 at 190^{th} epoch, without weight decay. We conduct experiments based on a small model with feature dimension D=64 and a large model with feature dimension D=128, termed as MacFormer-S and MacFormer-L, respectively. The inference latency denotes the time required to predict 32 agents concurrently on a single NVIDIA RTX 2080Ti GPU.

B. Ablation Study

We conduct ablation studies on the Argoverse 1 validation set using our 64-dimension MacFormer-S if not specified.

Table I. Importance of bilateral query on model performance.

Context fusion type	minFDE_6	minADE_6	MR_6	Latency	#Param
Bilateral query	**1.05**	**0.71**	**0.10**	**14ms**	**0.879M**
Stack attention [4]	1.08	0.73	0.11	83ms	2.756M

Effectiveness of Bilateral Query. By replacing bilateral query with stack attention [4] in our framework, we compare model performances of two context fusion manners. Specifically, we use 6 attention layers for stack attention (2 cross-attention and 4 self-attention) as the same as [4]. As reported in Table.I, bilateral query outperforms stack attention with 6x speed and using 68.1% fewer parameters. We validate the superiority of the proposed bilateral query in significantly reducing model parameters and inference latency.

Table II. Ablation studies on the proposed modules in our framework.

D	coupled map: relative motions	coupled map: map topology	bilateral query	reference extractor	minFDE_6	minADE_6	MR_6
64		✓	✓	✓	1.10	0.73	0.11
64	✓	✓		✓	1.15	0.75	0.13
64	✓	✓	✓		1.13	0.74	0.13
64	✓	✓	✓	✓	1.05	0.71	0.10
128	✓	✓	✓	✓	**0.98**	**0.67**	**0.08**

Importance of Each Module. As shown in Table.II, we illustrate the extent to which each module in our framework contributes to the prediction performance. First, relative motions in coupled map establish explicit connection of map and trajectory, thereby enhancing the map utilization. Second, using bilateral query allows effective context fusion, facilitating accurate reference extraction from coupled map. The absence of bilateral query results in a notable performance drop. Third, reference extractor has a significant impact on the performance, which provides map guidance for predictions. Without this module, the model cannot predict future trajectories using map-related references. The ablation of reference extractor is using agent motion feature to direct regress the predictions like implicit-fusion scheme instead of map-conditioned regression. Moreover, we qualitatively visualize the most interested centerline of each modality according to the score matrix in cross-attention of reference extractor. Fig.7 shows each modality in final predictions is accurately guided by its corresponding reference.

Table III. Ablation studies on multi-task optimization strategy (MTOS).

$\mathcal{L}_{primary}$	$\mathcal{L}_{couple}$	$\mathcal{L}_{capture}$	minFDE_6	minADE_6	MR_6
✓			1.14	0.77	0.13
✓		✓	1.09	0.73	0.11
✓	✓		1.11	0.74	0.10
✓	✓	✓	**1.05**	**0.71**	**0.10**

Ablation Studies on MTOS. We evaluated the effect of each task in MTOS as presented in Table.III. The coupled motion task $\mathcal{L}_{couple}$ can improve the performance since the map directly constrains predicted trajectories through optimizing future relative motions. On the other hand, excluding the motion capture task $\mathcal{L}_{capture}$ leads to inferior performance, highlighting that this supervision plays a critical role in reinforcing the capture of motion priors from historical trajectory.

C. Results

Comparison with State-of-the-art. We compare our method with the state-of-the-art models on Argoverse 1&2 and nuScenes benchmarks. Gray region indicates the official ranking metric. The single model results on Argoverse 1 is listed

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

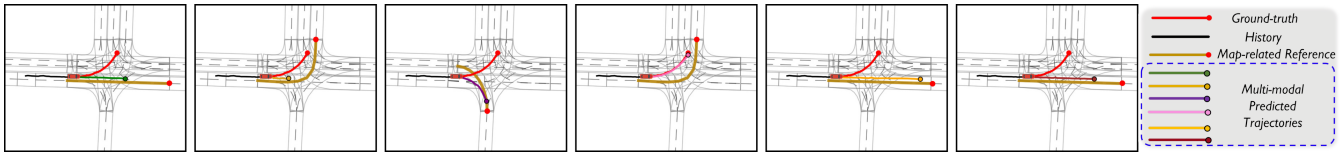

Fig. 7. Qualitative visualization of map-related references produced by **reference extractor**. (Sect.VI-B)

in Table.IV, where we also report the inference latency and model size using the official implementation if it exists.

Table IV. Single model comparisons on Argoverse 1 benchmark [18].

Method	brier-minFDE$_6$	brier-score	minFDE$_6$	minADE$_6$	MR$_6$	Latency	#Param
MacFormer-S	1.9021	0.6317	1.2704	0.8490	0.1311	**14ms**	**0.879M**
MacFormer-L	**1.8275**	**0.6115**	1.2160	0.8188	**0.1205**	19ms	2.485M
HiVT [9]	1.8422	0.6729	**1.1693**	**0.7735**	0.1267	69ms	2.529M
SceneTrans [4]	1.8868	0.6547	1.2321	0.8026	0.1255	257ms	15.296M
TPCN [7]	1.9286	0.6844	1.2442	0.8153	0.1333	-	-
DenseTNT [13]	1.9759	0.6944	1.2815	0.8817	0.1258	531ms	1.103M
mmTrans [6]	2.0328	0.6945	1.3383	0.8436	0.1540	129ms	2.607M
LaneGCN [3]	2.0539	0.6917	1.3622	0.8703	0.1620	173ms	3.701M

Compared to these methods, MacFormer-S achieves the competitive or better performance using 64.5% fewer parameters on average and with at least 10x lower latency. MacFormer-L outperforms all other methods listed in Table.IV in terms of the official ranking metric brier-minFDE$_6$ and MR$_6$ with the lowest latency (more than 4x) and fewest parameters. Also, MacFormer-L has the most reasonable probability prediction (best brier-score), which facilitates downstream decision-making and planning module for safe navigation. The above results demonstrate the superior prediction performance and real-time capability of our method.

Table V. Ensemble model comparisons on Argoverse 1 benchmark [18].

Method	brier-minFDE$_6$	brier-score	minFDE$_6$	minADE$_6$	MR$_6$	Latency	N
MacFormer-E	**1.7667**	**0.5526**	**1.2141**	0.8121	0.1272	**94ms**	5
MultiPath++ [5]	1.7932	0.5788	1.2144	**0.7897**	0.1324	738ms	5
HO+GO [14], [15]	1.8601	0.5682	1.2919	0.8904	**0.0846**	-	-

With the premise of ensuring the real-time requirement, we further improve the prediction performance using ensemble similar to [5]. The latency of ensemble model is calculated as $Nt_s + t_p$, where N represents the number of single models, t_s and t_p are latency of single model and post-process respectively. Our ensemble model, MacFormer-E, consists of 5 MacFormer-L with different random seeds. The results in Table.V show MacFormer-E outperforms other ensemble methods by a significant margin.

Table VI. Comparisons on Argoverse 2 benchmark [19].

Method	brier-minFDE$_6$	brier-score	minFDE$_6$	minADE$_6$	MR$_6$
MacFormer-E	**1.90**	**0.52**	1.38	**0.70**	0.19
GANet [17]	1.96	0.62	**1.34**	0.72	0.17
MTR [10]	1.98	0.54	1.44	0.73	**0.15**
THOMAS [16]	2.16	0.65	1.51	0.88	0.20

Comparisons between our best model with the state-of-the-art works on Argoverse 2 and nuScenes benchmarks is respectively listed in Table.VI and Table.VII. The results show our method significantly outperforms these models.

Table VII. Comparisons on nuScenes benchmark [20].

Method	minADE$_5$	minADE$_{10}$	minFDE$_1$
MacFormer-L	**1.21**	**0.89**	7.50
PGP [24]	1.27	0.94	7.17
THOMAS [16]	1.33	1.04	**6.71**
GOHOME [15]	1.42	1.15	6.99

Discussion on the utilization of map constraints. Practically, there do exist abnormal driving behaviors violating map constraints. If enforcing them as hard constraints, predicting abnormal behaviors is challenging. Thus, our method integrates constraints via encouragement, enabling predictions satisfying constraints while allowing abnormal behavior forecasting. Compared to constrained neural networks (enforcement manner), our method achieves more complete predicted behaviors and superior performance (Table.VIII).

Table VIII. Performance comparisons vs. constrained neural networks.

Method	Argoverse 1 [18]		nuScenes [20]	
	minFDE$_6$	minADE$_6$	minADE$_5$	minADE$_{10}$
MacFormer-L	**1.22**	**0.82**	**1.21**	**0.89**
PRIME [25]	1.56	1.22	-	-
GoalNet [26]	-	-	1.27	1.22

Qualitative Results. Fig.8 presents qualitative results of MacFormer-L on the Argoverse 1 validation set, including complicated intersections and long-range cases. Probability prediction results are also provided, with only the highest probability marked for clarity. The results demonstrate our model can predict accurate and feasible trajectories with reasonable and plausible multi-modality satisfying map constraints in complex scenarios. We further deployed our model in real-world scenarios using Argoverse tracking dataset [18] where we set B as the number of all agents in the scenario for joint prediction. The demonstration can be found at[1].

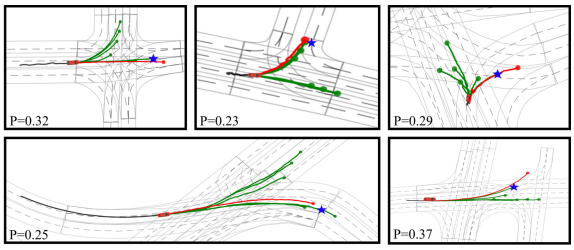

Fig. 8. Qualitative results of MacFormer-L. Historical trajectories is shown in black, ground-truth is shown in **red**, and predictions are shown in **green**. **Blue** star denotes the endpoint of the most confident predicted trajectory, where lower left corner shows its corresponding probability.

Robustness. Practically, history of agents sometimes contains invalid frames or noise, which poses a significant challenge to the robustness of trajectory prediction. We conclude three common cases: 1) new tracks with only a few frames, 2) missing frames in detection, and 3) noise in upstream processes.

[1]https://youtu.be/lCenh4XlH-4

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

To simulate these scenarios, we applied mask and random noise strategies to historical inputs and sent them to each prediction model on Argoverse 1 validation set. All models used are the 64-dimension version. Fig.9 shows that our method achieves superior performance compared to two state-of-the-art methods (LaneGCN [3] and MultiPath++ [5]) as the rate of invalid frames or standard deviation of noise increases. Our approach exhibits minimal performance degradation while maintaining optimal robustness.

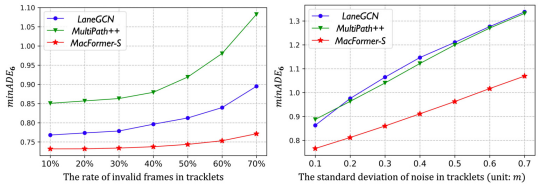

Fig. 9. The results of model robustness to imperfect tracklets.

Versatility. To showcase the versatility of our proposed framework, we conducted plugin-in experiments on Argoverse 1 validation set for classical models by directly replacing their map, context fusion, decoder, and output predictor with coupled map, bilateral query, reference extractor, and MTOS. All models used are the 64-dimension version. The results in Table.IX demonstrate that our framework effectively improves prediction performance while reducing model parameters by 24.1%, 56.2%, and 23.7% respectively for each component replaced. These findings highlight the versatile capability of our framework for trajectory prediction.

Table IX. The results of the versatility experiment on classical models.

Method	our framework	$minFDE_6$	$minADE_6$	MR_6	#Param
LaneGCN [3]		1.17	0.76	0.13	1.749M
	✓	**1.09**	**0.73**	**0.10**	**1.326M**
SceneTrans [4]		1.12	0.75	0.11	3.438M
	✓	**1.06**	**0.73**	**0.09**	**1.507M**
Multiath++ [5]		1.36	0.86	0.18	1.645M
	✓	**1.25**	**0.80**	**0.14**	**1.254M**

VII. Conclusion

We propose a real-time and robust trajectory prediction framework that effectively incorporates map constraints into the network. This is achieved by directly coupling map with trajectories and prediction conditioned on map-related references, respectively using coupled map and reference extractor. To promote learning map constraints, we present a multi-task optimization strategy (MTOS). Also, we develop a bilateral query scheme enhancing the computational efficiency for context fusion. Experiments show **MacFormer** achieves state-of-the-art performance on Argoverse 1&2, and nuScenes real-world benchmarks. Our model satisfies the requirements in terms of speed and model size, making it suitable for practical applications. Results also provide strong evidence for the robustness and versatility of our framework.

References

[1] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF CVPR*, 2020, pp. 11 525–11 533.

[2] S. Konev, K. Brodt, and A. Sanakoyeu, "Motioncnn: A strong baseline for motion prediction in autonomous driving," 2021.

[3] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *ECCV*. Springer, 2020, pp. 541–556.

[4] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, "Scene transformer: A unified architecture for predicting future trajectories of multiple agents," in *ICLR*, 2022.

[5] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, *et al.*, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 ICRA*. IEEE, 2022, pp. 7814–7821.

[6] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF CVPR*, 2021, pp. 7577–7586.

[7] M. Ye, T. Cao, and Q. Chen, "Tpcn: Temporal point cloud networks for motion forecasting," in *Proceedings of the IEEE/CVF CVPR*, 2021, pp. 11 318–11 327.

[8] L. Zhang, P. Li, J. Chen, and S. Shen, "Trajectory prediction with graph-based dual-scale context fusion," in *2022 IROS*. IEEE, 2022, pp. 11 374–11 381.

[9] 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 CVPR*, 2022, pp. 8823–8833.

[10] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *arXiv:2209.13508*, 2022.

[11] O. Makansi, J. Von Kügelgen, F. Locatello, P. Gehler, D. Janzing, T. Brox, and B. Schölkopf, "You mostly walk alone: Analyzing feature attribution in trajectory prediction," *arXiv:2110.05304*, 2021.

[12] 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," *arXiv:2008.08294*, 2020.

[13] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF ICCV*, 2021, pp. 15 303–15 312.

[14] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Home: Heatmap output for future motion estimation," in *2021 IEEE ITSC*. IEEE, 2021, pp. 500–507.

[15] ——, "Gohome: Graph-oriented heatmap output forfuture motion estimation," *arXiv e-prints*, pp. arXiv–2109, 2021.

[16] ——, "Thomas: Trajectory heatmap output with learned multi-agent sampling," *arXiv:2110.06607*, 2021.

[17] 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," *arXiv:2209.09723*, 2022.

[18] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF CVPR*, 2019, pp. 8748–8757.

[19] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[20] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF CVPR*, 2020, pp. 11 621–11 631.

[21] S. H. Park, G. Lee, J. Seo, M. Bhat, M. Kang, J. Francis, A. Jadhav, P. P. Liang, and L.-P. Morency, "Diverse and admissible trajectory forecasting through multimodal context understanding," in *ECCV*. Springer, 2020, pp. 282–298.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *CoRL*. PMLR, 2022, pp. 203–212.

[25] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, "Learning to predict vehicle trajectories with model-based planning," in *CoRL*. PMLR, 2022, pp. 1035–1045.

[26] L. Zhang, P.-H. Su, J. Hoang, G. C. Haynes, and M. Marchetti-Bowick, "Map-adaptive goal-based trajectory prediction," in *CoRL*. PMLR, 2021, pp. 1371–1383.