Implicit Scene Context-aware Interactive Trajectory Prediction for Autonomous Driving

Wenxing Lan, Dachuan Li, *Memeber, IEEE*, Qi Hao, *Memeber, IEEE*, Dezong Zhao, *Senior Member, IEEE*, and Bin Tian, *Memeber, IEEE*

***Abstract*—The accurate prediction of behaviors of surrounding traffic participants is critical for autonomous vehicles (AV). How to fully encode both explicit (e.g., map structure and road geometry) and implicit scene context information (e.g., traffic rules) within complex scenarios is still challenging. In this work, we propose an implicit scene context-aware trajectory prediction framework (the PRISC-Net, Prediction with Implicit Scene Context) for accurate and interactive behavior forecasting. The novelty of the proposed approach includes: 1) development of a behavior prediction framework that takes advantage of both model- and learning-based approaches to fully encode scene context information while modeling complex interactions; 2) development of a candidate path target predictor that utilizes explicit and implicit scene context information for candidate path target prediction, along with a motion planning-based generator that generates kinematic feasible candidate trajectories; 3) integration of the proposed target predictor and trajectory generator with a learning-based evaluator to capture complex agent-agent and agent-scene interactions and output accurate predictions. Experiment results based on vehicle behavior datasets and real-world road tests show that the proposed approaches outperform state-of-the-art methods in terms of prediction accuracy and scene context compliance.**

***Index Terms*—Trajectory Prediction, Interaction, Semantic Context, Traffic Rules**

I. Introduction

ACCURATELY predicting the intention and behavior of surrounding traffic agents is critical for autonomous vehicles (AVs) [1], [2], which is premise for reasonable decision-making as well as safe motion planning [3]–[8]. In complex traffic scenarios, the behavior of an agent is usually shaped by various agent-agent interactions and complex scene context (including both explicit and implicit ones). The explicit scene context refers to environment information that can be directly represented using certain map formats (e.g. road network geometry, road boundaries, etc.). In addition, an AV is also required to infer implicit scene context from formatted data (e.g. lane change is only allowed at solid lane markings). Therefore, it is challenging to accurately predict the interactive motions of traffic agents, whose intentions and behaviors are multimodal and affected by various factors [4].

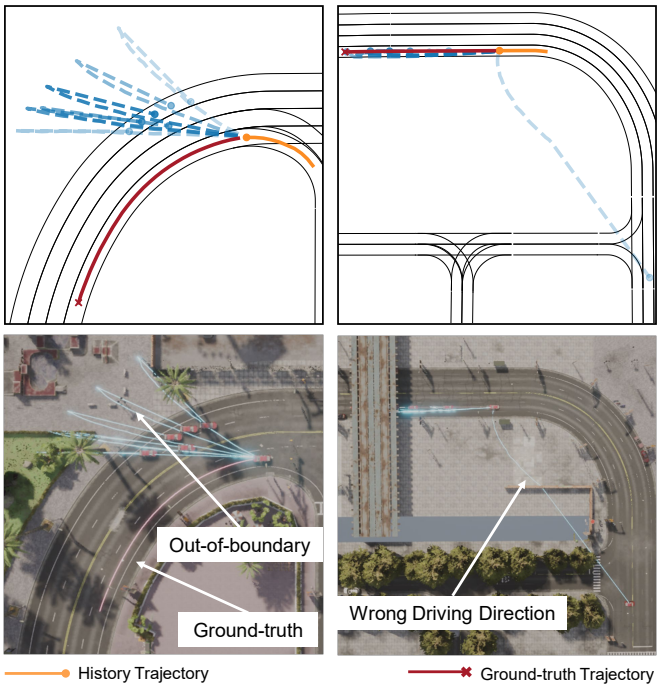


Fig. 1. Examples of incorrect behavior predictions due to ignorance of implicit scene context. *Left*: Out-of-road boundary predictions. *Right*: Predictions with incorrect driving direction. Dots represent the final position of vehicles.

Existing behavior prediction frameworks can be divided into two categories: model-based and learning-based approaches. The former forecasts agents' future behavior using explicit models that describe agents' physical motion states [5]. However, such approaches rely on the assumption of simple and basic kinematics, and they are typically incapable of addressing the multi-modality and interaction factors in complex scenarios. In recent years, learning-based approaches have been widely applied in behavior prediction. Such approaches typically utilize deep neural network (DNN) frameworks (e.g., convolutional neural network (CNN) [9], graph neural network (GNN) [10]) to extract interaction and scene context features from certain representations of map and agents' states [4], [5], [10]–[17]. Using these scene context features, such approaches directly regress future trajectories of the target agent based

This work was supported in part by the National Natural Science Foundation of China under Grants 52272419 and 62261160654, and in part by the National Key Research and Development Program of China under Grant 2022YFB4703700. (*Corresponding authors: Dachuan Li; Qi Hao.*)

W. Lan, D. Li and Q. Hao are with Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, 518055 Shenzhen, China, and are also with the Department of Computer Science and Engineering, Southern University of Science and Technology, 518055 Shenzhen, China (e-mail: 12032882@mail.sustech.edu.cn, dachuanli86@gmail.com, haoq@sustech.edu.cn)

D. Zhao is with James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, United Kingdom (e-mail: dezong.zhao@glasgow.ac.uk)

B. Tian is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, and with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: bin.tian@ia.ac.cn)

in an end-to-end manner [10], [15], [16]. Although learning-based approaches show promising prediction accuracy and effectiveness in modeling interactions, they typically ignore the feasibility (i.e., vehicle kinematics) and semantic constraints (i.e., map structure and road geometry). Some recent research efforts attempt to address the kinematic feasibility by incorporating model-based motion planning to generate feasible candidate trajectories [13], [17], [18], but they cannot fully account for implicit scene context constraints such as traffic rules.

Despite the achievements of the existing approaches, developing an accurate and reliable behavior prediction framework still faces the following challenges: 1) *Fully encoding the scene context information*. The agents' motions are shaped by explicit (e.g., road structures) and implicit scene context factors (e.g., traffic rules). How to fully incorporate and represent such scene context knowledge is the key to ensuring prediction accuracy. 2) *Guaranteeing the feasibility of predicted behaviors*. Most existing learning-based approaches follow a model-free paradigm without considering the kinematic constraints of vehicles. It is necessary to ensure the predicted trajectories are physically feasible for AVs. 3) *Modeling the complex interactions*. Capturing the interactions among agents and the environment is critical to achieving reliable behavior prediction in complex scenarios. How to effectively represent multi-agent and agent-to-environment interactions is still a challenging issue.

To address these issues, we propose the Prediction with Implicit Scene Context (*PRISC-Net*), a novel framework that takes advantage of both model- and learning-based approaches to provide accurate and scene context-compliant trajectory predictions. The proposed model-based candidate trajectory generation stage encodes scene context and kinematic constraints, while the learning-based evaluation stage copes with complex interactions and multi-modality. The major contributions of this paper include:

- Developing an implicit scene context-enhanced candidate path target predictor. It fully incorporates implicit and explicit scene context information to generate candidate target waypoints and ensures scene context compliance.
- Developing a novel planning-based trajectory generator that provides candidate trajectories with kinematic feasibility guarantees.
- Integrating the proposed generator with a learning-based evaluator to encode complex multi-agent and agent-to-environment interactions and output final trajectory predictions.
- Conducting experiments based on vehicle behavior datasets and real-world road test. The proposed approaches outperform existing methods in terms of traffic-rule compliance and accuracy (Source codes are available at: https://github.com/Joe12138/PRISC-Net-V1).

The remainder of the paper is structured as follows. Section II reviews related work. Section III formulates the problem and provides the overview of the proposed framework. Section V presents the proposed methods in detail. Section VI compares our approach with several state-of-the-art approaches on both real-world and simulation datasets. Section VII concludes this paper and discusses future work.

II. Related Work

A. Scene Context Encoding

For AV trajectory prediction applications, rich scene context features are required to be learned from the elements of the traffic scene, including the map context and historical state of agents. Rich scene context can be provided by datasets collected in the real-world (e.g., INTERACTION [19], Argoverse [20]) and simulators (e.g., CARLA [21], MetaDrive [22]). Furthermore, some existing works [23]–[27] apply Scenarios Engineering (SE) to generate rich scene context for algorithm training and testing automatically. For example, Guo *et al.* calibrate trajectory prediction through SE to improve the evaluation index of prediction by utilizing more traffic information and attribute characteristics [23]. Li *et al.* apply computer graphics (CG) to clone real highway scenarios and generate synthetic multi-challenge video datasets, which can test foreground detection algorithms after translating [24]. The scene context encoding methods can be divided into two categories: rasterized encoding [15], [28]–[32] and vectorized encoding [4], [10]–[12], [33].

Rasterized encoding methods first extract map elements (e.g., lane boundaries, traffic lights, crosswalks) from the high definition (HD) map, then render these scene elements in different colors or masks in bird's eyes view RGB images, and finally use the convolutional neural network (CNN) [9] to encode the image. Based on this encoding method, both Cui *et al.* and Djuric *et al.* employ a CNN to extract scene context features from rasterized images [15], [29]. Similarly, MultiPath [28] uses CNNs to extract agent-agent interaction, scene, and agent features from top-down rasterized images. Hong *et al.* encodes the scene context by using a CNN backbone of 2D convolutions, whose input is top-down rasterized images [30]. Heatmap output for future motion estimation (HOME) [31] rasterizes the HD map in 5 semantic channels, then applies a classic CNN model to encode scene context. A CNN is designed in Heterogeneous edge-enhanced graph attention network (HEAT-I-R) [32] to extract road features from a bird's eye view of the driving scene. These rasterized methods cannot capture the structural information of HD maps and do not allow non-grid sampling of goal points due to the shape of convolutions [12]. Furthermore, employing CNN makes computation become expensive [4].

In contrast to rasterized encoding method, vectorized encoding methods abstract all geographic entities (e.g., roads, traffic lines) and traffic agents as polylines. Therefore, the structural features of scene context are better captured by vectorized encoding methods [12]. Neural network with vector representation (VectorNet) [10], [23], target-driven trajectory prediction (TNT) [11] and target-driven trajectory prediction with dense goal set (DenseTNT) [12] all use a multilayer perceptron (MLP) [34] to learn object features from vectorized polylines, then employs a graph neural network to extract high-order interactions based on learned object features. Nonetheless, these methods fail to consider the relations between

objects and cannot learn any semantic information about traffic rules (e.g., traffic signs). Lane graph convolutional network (LaneGCN) [33] encodes the driving scene as a lane graph and applies graph convolutions with adjacency matrices to capture the complex scene feature from the lane graph. Hierarchical vector transformer (HiVT) [4] employs hierarchical vector transformer to learn multi-agent and agent-scene interaction from vectorized scenes. However, all of the methods cannot fully encode the scene context information since they do not consider implicit scene context (e.g., traffic signs, speed limit) at all when encoding the driving scene.

Encoding scene context as vectors is beneficial for providing more accurate and detailed information to trajectory predictors. However, most existing vectorized encoding methods only utilize explicit scene context (e.g., road structures), but they typically ignore implicit scene context (e.g., traffic rules) [5]. Therefore, such approaches may lead to inaccurate and unreasonable final predictions. In this work, we propose a candidate path target predictor that utilizes both explicit and implicit scene context by enhancing vectorized representations with a rule-constrained path search, respectively (More details are presented in Section III C).

B. Candidate Trajectory Generation

Since the behavior of agents is uncertain and multi-modal, trajectory predictors typically generate several possible candidate trajectories for further selection. Therefore, the quality of candidate trajectories significantly affects the accuracy and feasibility of the final predictions. Existing candidate trajectory generation methods can be divided into two categories: neural network regression-based methods [4], [10]–[12], [31] and model-based methods [13], [17], [18].

Neural network regression methods apply neural networks (e.g., MLP, long short-term memory (LSTM) [35]) to regression predicted trajectories based on scene context features or interaction features. VectorNet [10], TNT [11], HiVT [4] and DenseTNT [12] all employ MLP to regress future trajectories of the target vehicle based on scene features learned from vectorized scene context. A convolutional decoder is adopted by HOME to output an image with sampled target locations; then, a separate model is applied to generate full trajectories connecting the initial agent position to all sampled locations on the image [31]. HEAT-I-R [32] applies LSTM to generate future trajectory based on scene context and interaction features. Similarly, both Li *et al.* and Kaouther *et al.* use LSTM to generate future trajectory based on interaction features [36], [37]. All of these regression methods can generate highly accurate but unreasonable and kinematic unfeasible future trajectories of the target vehicles, as shown in Fig. 1 and 9.

To account for feasibility constraints in behavior prediction, model-based motion planners are integrated into the prediction framework to generate candidate trajectories in some recent model-based methods [13], [17], [18]. The Frenét [38] and polynomial curve-based planners [39], such as quintic polynomial planner, are used in [17] and [13], [18] to generate possible future trajectories, respectively. In contrast to those trajectories directly regressed by neural networks, those trajectories generated by model-based planners can inherently satisfy kinematic constraints with guaranteed feasibility for AVs. However, the Frenét planner relies on scenario-specific parameters and it is thus sensitive to the input reference line, making it less suitable for many prediction tasks (Fig. 6). Therefore, trajectories generated by the Frenét planner may enter an unreasonable area, as shown in Fig. 7. Moreover, the quintic polynomial planner-based prediction scheme requires additional vehicle states (e.g., x-y coordinate, speed, acceleration, and heading), which is difficult to obtain or predict yet. The main flaw of planning-based approaches lies in their difficulties in utilizing implicit scene context information. Compared to regression methods, trajectories generated by model-based methods are more reasonable and kinematic feasible.

However, how to overcome the inherent limitations of existing curve-based planners and encode implicit scene context information (e.g., traffic rules) in trajectory generation are still challenging. In this work, we propose an optimization-based parameter-free planner without requiring additional vehicle state observations and reference lines. In addition, implicit scene context information is utilized to constrain the final position of future trajectories (c.f. Section III D).

C. Interaction Modeling

Modeling multi-agent and agent-scene interaction is crucial for the real-world application of trajectory predictors in interactive scenarios. Deo and Trivedi employ convolutional social pooling to learn inter-agent interaction from an occupancy grid [40]. HOME [31] uses attention [41] to model agent interaction by generating a query vector and key as well as value vectors for the target agent and other actors, respectively. Similarly, both Li *et al.* and Kaouther *et al.* apply multi-head attention to model higher-order interactions rather than pairwise vehicle interactions [36], [37]. A graph neural network is adopted by VectorNet [10], TNT [11], and DenseTNT [12] to model high-order multi-agent and agent-scene interactions. LaneGCN [33] proposes a FusionNet to capture a complete set of actor-map interactions. HiVT [4] employs a transformer to model local inter-agent and agent-scene interactions in a region area; then, global interaction is modeled by another transformer. HEAT-I-R [32] proposes a heterogeneous edge-enhanced graph attention network to extract inter-agent interaction.

Interaction features should include the degree to which other traffic entities influence the target vehicle. Therefore, attention [41] is more suitable for modeling interactions. However, most existing works ignore the interaction among future trajectories of the target vehicle when modeling interactions, making them unable to predict multi-modal behaviors. In this work, we employ various self-attention models to capture multi-agent, agent-scene, and future agent-agent interactions. Using these interaction features, the proposed framework can effectively cope with the multi-modality of interactive behaviors.

III. Problem Formulation

Denoting the autonomous ego-vehicle as $\mathbf{v}_{\text{ego}}$, and the observed state $\mathcal{S}$ of its surrounding vehicles $\mathcal{V}$ ($\mathbf{v}_{\text{ego}} \notin \mathcal{V}$)

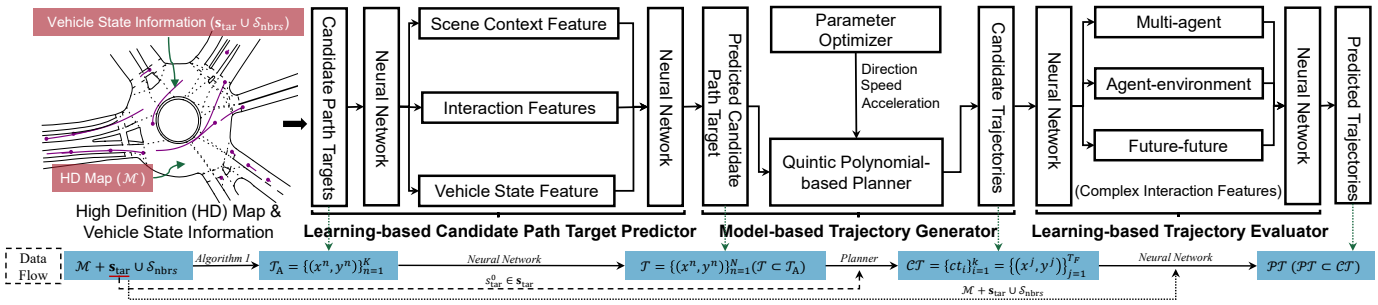


Fig. 2. Overview of the proposed implicit scene context-enhanced trajectory prediction framework (PRISC-Net). 1) The *learning-based candidate path target predictor* predicts future candidate path targets $\mathcal{PT}$ of the target vehicle, using HD map $\mathcal{M}$ and vehicle's state observations $\{\mathbf{s}_{\text{tar}} \cup \mathcal{S}_{\text{nbrs}}\}$ (cf. Section V-A) as the input. 2) The *model-based trajectory generator* generates kinematic feasible candidate trajectories $\mathcal{CT}$ starting from the initial position of a target vehicle based on the predicted candidate path targets $\mathcal{PT}$ (cf. Section V-B). 3) The *learning-based trajectory evaluator* evaluates candidate trajectories and rank them by $\mathcal{E}$ with complex interaction features, and the most possible predicted trajectories $\mathcal{T}_{\text{tar}}$ are outputted. (cf. Section V-C).

can be obtained by the detection-and-tracking modules of $\mathbf{v}_{\text{ego}}$. In addition, we assume that $\mathbf{v}_{\text{ego}}$ has access to a pre-built high definition (HD) map $\mathcal{M}$ (including explicit and implicit scene context information) of the current traffic scenario to obtain lane connectivity, traffic rules, and other semantic information. For a given target vehicle $\mathbf{v}_{\text{tar}} \in \mathcal{V}$ for prediction, we denote its surrounding vehicles as $\mathcal{V}_{\text{nbrs}} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m\}$ along with their state sequences as $\mathbf{s}_{\text{i}} = \{s_i^{-T_H+1}, s_i^{-T_H+2}, ..., s_i^0\} (i \in \{1, 2, ..., m\})$ (where T_H denotes the history observation time horizon, and each state vector is composed of the position (x, y), heading θ and velocity v). Therefore, the set of observed state of $\mathcal{V}_{\text{nbrs}}$ is represented by $\mathcal{S}_{\text{nbrs}} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_m\}$, where $m = |\mathbf{V} \setminus \{\mathbf{v}_{\text{tar}}\} \bigcup \{\mathbf{v}_{\text{ego}}\}|$ ($|\cdot|$ presents the number of elements in a set). $\mathbf{s}_{\text{tar}} = \{s_{\text{tar}}^{-T_H+1}, s_{\text{tar}}^{-T_H+2}, ..., s_{\text{tar}}^0\}$ denotes the historical states of the target vehicle $\mathbf{v}_{\text{tar}}$.

Given $\mathbf{v}_{\text{tar}}$, $\mathcal{V}$, $\mathcal{M}$, $\mathbf{s}_{\text{tar}}$ and $\mathcal{S}_{\text{nbrs}}$, the objective of the proposed framework is to predict the possible future trajectories $\mathcal{T}_{\text{tar}}$ of $\mathbf{v}_{\text{tar}}$, which consists of states of $\mathbf{v}_{\text{tar}}$ up to the prediction horizon T_F. In addition, each predicted trajectory in $\mathcal{T}_{\text{tar}}$ should satisfy feasibility constraints $\mathcal{C}$ consisting of scene context constraints $\mathcal{C}_E$ and kinematic constraints $\mathcal{C}_K$.

IV. Framework of the Proposed PRISC-Net

The overall framework of the proposed PRISC-Net is shown in Fig. 2, and it consists of three primary components: *candidate path target predictor*, the *trajectory generator* and the *trajectory evaluator*.

- *Candidate Path Target Predictor (learning-based):* The candidate path target predictor $\mathcal{P}$ aims to predict a set of candidate path target (i.e., final key position ($x-y$ coordinate, denoted as $\mathcal{PT} = \{\tau^n\}_{n=1}^N = \{(x^n, y^n)\}_{n=1}^N$) along the possible path) of the given target vehicle at certain time step in the future, using features learned by a neural network. The input of $\mathcal{P}$ includes HD map $\mathcal{M}$ and state information (i.e., $x-y$ coordinate, heading and velocity) of vehicles $\{\mathbf{s}_{\text{tar}} \cup \mathcal{S}_{\text{nbrs}}\}$. Firstly, $\mathcal{P}$ applies the candidate path search algorithm (Algorithm 1) to search possible reachable paths of the given target vehicle with HD map and vehicles' state information. Then, several candidate final positions of the given target vehicle ($\mathcal{PT}_{\text{A}} = \{\tau^n\}_{n=1}^K = \{(x^n, y^n)\}_{n=1}^K$) are sampled from the centerline of possible reachable paths with a uniform distance. Finally, $\mathcal{P}$ selects part of candidate final positions $\mathcal{PT}$ ($\mathcal{PT} \subset \mathcal{PT}_{\text{A}}$) as candidate path targets using features learned by a neural network from the HD map and vehicles' state information (More details can be found in Section V-A, and the overview of $\mathcal{P}$ is shown in Fig. 3.).
- *Feasible Candidate Trajectory Generator (model-based):* Given the predicted candidate path targets τ^i ($\tau^i \in \mathcal{T}$), the feasible candidate trajectory generator $\mathcal{G}$ generates kinematic feasible candidate trajectories $\mathcal{CT} = \{ct_i\}_{i=1}^k$ ($ct_i = \{(x^j, y^j)\}_{j=1}^{T_F}$ where T_F is the prediction time horizon, and k denotes the number of candidate trajectories) staring from the initial position of a target vehicle. Concretely, $\mathcal{G}$ takes the initial state of the given target vehicle s_{tar}^0 ($s_{\text{tar}}^0 \in \mathbf{s}_{\text{tar}}$) and the predicted candidate path target τ^i ($\tau^i \in \mathcal{PT}$) as inputs.
- *Trajectory Evaluator (learning-based):* The predicted candidate trajectories $\mathcal{CT}$ are then evaluated and ranked by the learning-based trajectory evaluator $\mathcal{E}$. The most possible trajectory $\mathcal{T}_{\text{tar}}$ ($\mathcal{T}_{\text{tar}} \subset \mathcal{CT}$) is finally selected and outputted by $\mathcal{E}$, using implicit multi-agent and agent-to environment interactions captured by the neural network-based evaluator.

V. Proposed Methods

A. Scene Context-constrained Learning-based Candidate Path Target Predictor

The proposed candidate path target prediction pipeline (Fig. 3) consists of 3 stages: candidate target sampling (cf. Section V-A1), scene feature extraction (cf. Section V-A2) and state feature encoding (cf. Section V-A3). Firstly, to account for both the explicit and implicit scene context information, a traffic rule-constrained path search algorithm (cf. Algorithm 1) is applied to search reachable paths from the HD map, and then the candidate final position of the given target vehicle is sampled from these paths. Secondly, VectorNet [10] and LSTM [35] are employed to extract scene context features from the vectorization of traffic scene, as well as the trajectories of surrounding agents and target vehicle features from the state information of the target vehicle, respectively. Thirdly,

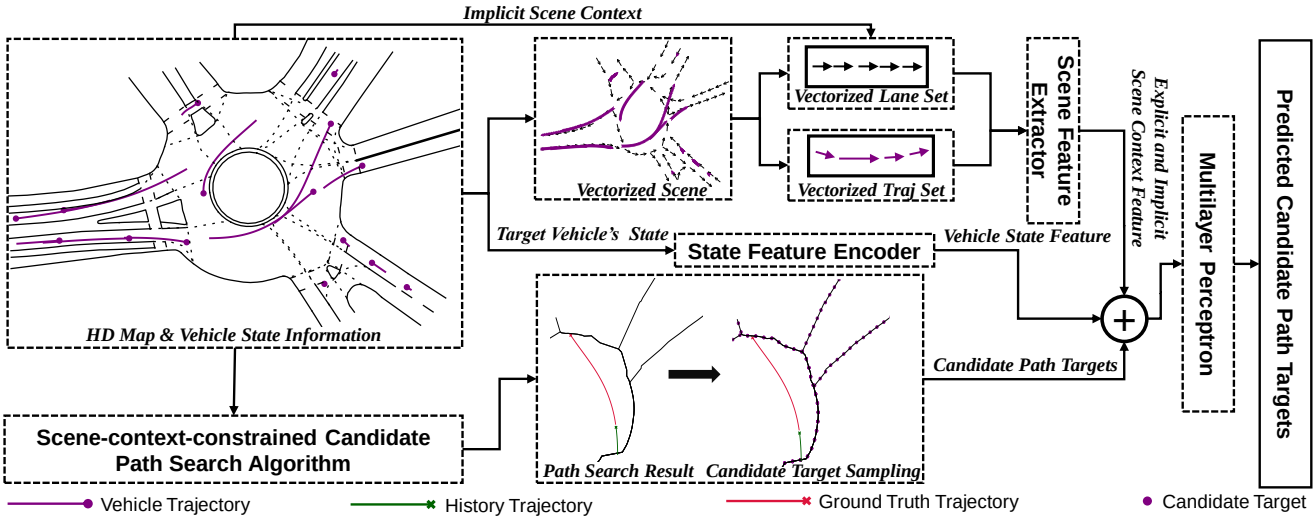


Fig. 3. Pipeline of the proposed candidate path target predictor. Firstly, an implicit scene context-constrained path search algorithm (cf. Algorithm 1) is applied to search reachable paths, and candidate targets are sampled along these paths. Secondly, VectorNet [10] and LSTM [35] are employed to extract scene context features from the vectorized traffic scene and target vehicle feature from its state observations, respectively. Finally, the final candidate path targets and their probabilities are predicted according to scene and state features.

the final candidate path targets and their probabilities are predicted according to features extracted in the second stage.

1) Candidate Path Target Sampling: As the vehicles are supposed to follow traffic regulations and drive only in the drivable area of the road, reachable paths $\mathcal{RP} = \{rp_i\}_{i=1}^{M}$ ($rp_i = \{(x^j, y^j)\}_{j=1}^{PL}$ where rp_i is centerline of path rp_i[1]) of the target vehicle are firstly searched from the road network under traffic rule constraints (e.g., road geometry, speed limit). Our path search algorithm is implemented using the Depth-First-Search (DFS) [42] on the HD map (Algorithm 1). To account for the constraints of implicit scene context (e.g., traffic rules), we incorporate speed limit and traffic sign information to restrict the search for drivable paths. Therefore, reachable paths generated by the proposed approach are more scene context-compatible, and the efficiency is also improved as the search space is restricted. Assuming that $\mathbf{v}_{\text{tar}}$ normally does not deviate too much from lanes, the candidate path target ($\mathcal{T}_{\text{A}} = \{\tau^n\}_{n=1}^{K} = \{(x^n, y^n)\}_{n=1}^{K}$), where (x, y) denotes a 2-D location in the global map) are uniformly sampled along the centerline of the generated drivable paths (marked as purple solid dots in Fig. 3). The effectiveness of this module is shown in Section VI-C1.

2) Scene Feature Extraction: To efficiently encode the scene context information, we use the hierarchical graph neural network-based VectorNet [10]–[12] to extract scenario context features from HD map and state observation of the vehicles. Please note that compared to the original VectorNet [10], we add extra implicit scene context (e.g., speed limit, traffic sign information) into lane node features, and implementation details can refer our codebase, which is available at https://github.com/Joe12138/PRISC-Net-V1). Taking advantage of the VectorNet, the proposed framework learns features from vectorized center lines of road networks and trajectories of agents in the region of interest. In this manner, we can fully encode the structured scene context information and implicit multi-agent, agent-scene interactions as a unified vector representation of features.

Algorithm 1: Scene-context-constrained Candidate Path Search

Input: HD Map: $\mathcal{M}$; historical state sequence of target vehicle: $\mathbf{s}_{\text{tar}} = \{s_{\text{tar}}^{-T_H+1}, s_{\text{tar}}^{-T_H+2}, ..., s_{\text{tar}}^{0}\}$
Output: Centerline waypoints of drivabel paths: $CL = \{cl_1, cl_2, ..., cl_m\}$

$\mathcal{M}_{\text{LC}} \leftarrow$ Extract lane connectivity from $\mathcal{M}$;
$\mathcal{SL} \leftarrow$ Extract speed limits from $\mathcal{M}$;
$\mathcal{TS} \leftarrow$ Extract traffic signs from $\mathcal{M}$;
$\mathcal{L}_{\text{lane}} \leftarrow$ Find lanes where the target vehicle is on according to $\mathcal{M}$ and $\mathbf{s}_{\text{tar}}$;
$CL \leftarrow$ Empty set;
for $lane \in \mathcal{L}_{\text{lane}}$ **do**
  $\mathcal{P} \leftarrow$ Apply DFS [42] on $\mathcal{M}_{\text{LC}}$;
  for $path \in \mathcal{P}$ **do**
    $path_{\text{SL}} \leftarrow$ Cut off $path$ with $\mathcal{SL}$ and $\mathcal{TS}$;
    Add $path_{\text{SL}}$ to CL;
  end
end
$CL \leftarrow$ Filter CL with end point position;

3) State Feature Encoding: Candidate target prediction is essentially a multi-label task that requires intensive features to guarantee the prediction accuracy. Therefore, extra state features of $\mathbf{v}_{\text{tar}}$ are extracted from its historical state sequence (including x-y coordinate, velocity, and heading). Furthermore, the proposed state feature extractor is structured with a temporal convolutional layer [43] followed by a long short-term memory (LSTM) layer [35]. The effectiveness of this module is shown in Section VI-C2 and Fig. 13.

[1] rp_i is usually represented with discrete points, sampled with uniform distance interval σ. Thus, $PL = \lceil \frac{l_{rp_i}}{\sigma} \rceil$, where l_{rp_i} is the length of rp_i and $\lceil \cdot \rceil$ is ceiling function.

Using the sampled candidate targets ($\mathcal{PT}_{\mathrm{A}} = \{\tau^n\}_{n=1}^{K} = \{(x^n, y^n)\}_{n=1}^{K}$), the extracted scene feature ($\mathcal{F}_{\mathrm{sce}}$) and state feature ($\mathcal{F}_{\mathrm{sta}}$) as the input, a 3-layer multilayer perceptron (MLP) [34] (namely M_{p}) is trained to predict the likelihood that a candidate target is the possible position of $\mathbf{v}_{\mathrm{tar}}$ in a prediction horizon. Since the candidate targets are sampled exactly from the centerlines of lanes, We utilize an additional MLP M_{d} to regress the distance between a candidate target and the ground truth position. Identical to M_{p}, M_{d} is also a 3-layer MLP. The loss function used for training candidate target predictor is as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{p}}(\pi, \mu) + \mathcal{L}_{\mathrm{d}}(M_{\mathrm{d}}(x), M_{\mathrm{d}}(y), \triangle x^{\mu}, \triangle y^{\mu}) \quad (1)$$

where

$$\pi(\tau^n|(\mathcal{F}_{\mathrm{sce}}, \mathcal{F}_{\mathrm{sta}})) = \frac{\exp\{M_{\mathrm{p}}(\tau^n, (\mathcal{F}_{\mathrm{sce}}, \mathcal{F}_{\mathrm{sta}}))\}}{\sum_{\tau'} \exp\{M_{\mathrm{p}}(\tau', (\mathcal{F}_{\mathrm{sce}}, \mathcal{F}_{\mathrm{sta}}))\}} \quad (2)$$

is a discrete distribution over the candidate positions τ^n, and $\mathcal{L}_{\mathrm{p}}$ and $\mathcal{L}_{\mathrm{d}}$ are cross entropy and Huber loss, respectively. μ is the candidate target closest to the ground truth position μ_{gt}, and $\triangle x^{\mu}$ and $\triangle y^{\mu}$ are distance between μ and μ_{gt} in x and y direction, respectively.

B. Planning-based Feasible Candidate Trajectory Generator

Given the reference reachable path waypoints, we propose an optimization-based planner to generate kinematic-feasible and scene context-compliant candidate trajectories $\mathcal{CT} = \{ct_i\}_{i=1}^{k}$ ($ct_i = \{(x^j, y^j)\}_{j=1}^{T_F}$ where T_F is prediction horizon, and k denotes the number of candidate trajectories). The proposed PRISC-Net framework adopts polynomial curve-based planning to guarantee the smoothness and feasibility of generated trajectories.

We adopt the quintic polynomial planner to generate feasible and smooth trajectories connecting a given initial state (i.e., the final observed position of $\mathbf{v}_{\mathrm{tar}}$ in the previous prediction interval) to a goal state (candidate reference waypoints provided by the candidate target predictor). However, the quintic polynomial planner requires additional input parameters (i.e., velocity, acceleration, and heading at the key points) not provided by the candidate target predictor. In our proposed trajectory generator, such input parameters are determined by solving the following optimization problem:

$$\arg \min_{v_{\mathrm{tar}}, a_{\mathrm{tar}}, \theta_{\mathrm{tar}}} (k_j \sum_{i=1}^{10T_{\mathrm{Pred}}} J_i) + k_v \triangle v + k_s \triangle s_{\mathrm{Lat}} \quad (3)$$

$$s.t. \max(0, v_{\mathrm{start}} - \alpha_1) \leq v_{\mathrm{tar}} \leq \min(v_{\mathrm{start}} + \alpha_1, SL) \quad (4)$$

$$\max(-2, a_{\mathrm{start}} - \alpha_2) \leq a_{\mathrm{tar}} \leq \min(a_{\mathrm{start}} + \alpha_2, 3) \quad (5)$$

$$\max(-\pi, \theta_{\mathrm{start}} - \alpha_3) \leq \theta_{\mathrm{tar}} \leq \min(\theta_{\mathrm{start}} + \alpha_3, \pi) \quad (6)$$

where J_i, $\triangle v$, and $\triangle s_{\mathrm{Lat}}$ are the jerk at time step i, the velocity difference and lateral offset in the whole predicted trajectory, respectively. α_1, α_2, α_3, k_j, k_v and k_s are all positive coefficients for adjusting the range limits (The value of α_1, α_2 and α_3 can be determined according to the physical properties of a general vehicle model. The setting of k_j, k_v and k_s used in Song *et al* [17] is adopted in this work). Therefore, given x-y coordinates $(x_{\mathrm{start}}, y_{\mathrm{start}})$, velocity v_{start}, acceleration a_{start} and the orientation of driving θ_{start} of the initial state, as well as x-y coordinates $(x_{\mathrm{tar}}, y_{\mathrm{tar}})$ of target state and speed limit SL at a given target state, the input parameters of the polynomial planner can be determined by minimizing the objective function given by Eq. 3. In addition, the optimization problem can be solved using an off-the-shelf solver, i.e., SciPy package [2], or using the sampling-based method used by Song *et al.* [17]. The effectiveness of the trajectory generator is shown in Section VI-C3.

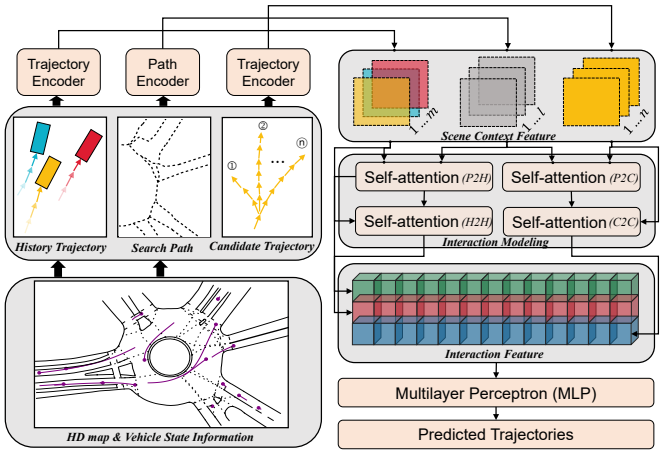


Fig. 4. Pipeline of the proposed learning-based trajectory evaluator. Firstly, historical and predicted candidate trajectories and reachable paths are encoded as scene context features (cf. Section V-C1). Secondly, four self-attention modules are employed to extract multi-agent and agent-scene interactions (cf. Section V-C2). Finally, a multi-layer perceptron (MLP) is used to evaluate each predicted candidate trajectory according to interaction features and output the ultimate trajectory prediction.

C. Interaction-aware Learning-based Trajectory Evaluation

As the trajectory generation stage yields multiple possible trajectories of $\mathbf{v}_{\mathrm{tar}}$, a learning-based evaluator is utilized to determine the ultimate trajectory prediction result by scoring all possible candidates. The trajectory evaluation is essentially a multi-label task, and the evaluator needs to select the most possible one or several trajectories from the candidates by considering various factors, such as agent-scenario interactions and scene context. Furthermore, the proposed PRISC-Net extracts such implicit interaction features (cf. Section V-C2) using the Self-attention [41] mechanism, with the encoded state observations of traffic entities, the scene context information, and the predicted candidate trajectories as the input.

1) Trajectory and Path Encoding: Before capturing interactions among the traffic entities, we first encode observed states of traffic entities, the reachable paths, and predicted candidate trajectories of $\mathbf{v}_{\mathrm{tar}}$ in the scene context. Each observed state sequence and predicted trajectory are discretized as a sequence of 2-D positions with equal time intervals, and each reachable path is split with the same distance interval. To better model the relative motion of $\mathbf{v}_{\mathrm{tar}}$ and its reference path, the Frenét coordinate is adopted in addition to the Cartesian coordinate to form a combined spatial representation. These paths and the historical and predicted trajectories are then encoded with

[2]More details are available at https://scipy.org/

a temporal convolution layer [43] followed by a long short-term memory (LSTM) layer [35]. The trajectory encoder uses a unidirectional LSTM [35], while the paths encoder employs a bidirectional LSTM [35] since the predicted path waypoints are not associated with direction information.

2) Modeling Interactions : Similar to [17], to fully capture the implicit interactions between scene context factors and dynamic agents, four Self-attention [41] modules are utilized to extract path-to-historical trajectory (P2H), historical trajectory-to-historical trajectory (H2H), path-to-candidate prediction (P2C), and candidate prediction-to-candidate prediction (C2C) interaction features, respectively. Concretely, the abstract spatial relationship between traffic entities and the roads is considered as agent-scenario interactions, which are classified into historical and future interactions. The historical one is extracted by the P2H module,by encoding reachable paths with historical state observations of traffic entities. The future one is encoded by the P2C which interprets the dependency between reachable paths and candidate future trajectories of the target vehicle. On the other hand, agent-agent interactions represent the spatial-temporal interrelationship of the behaviors of the traffic entities. Such interactions are captured by H2H and in the past time domain, using the trajectory encoding of agents and encoded historical agent-scenario interaction as inputs. Finally, the C2C interprets the differences between candidate trajectories. These interaction features are encoded as high-dimension vectors, which are concatenated to fully describe the future trajectories. The Self-attention mechanism is formulated as follows:

$$\mathbf{Q}_i = \mathbf{W}^{\mathrm{Q}} a_i, \mathbf{K}_i = \mathbf{W}^{\mathrm{K}} b_i, \mathbf{V}_i = \mathbf{W}^{\mathrm{V}} c_i \tag{7}$$

$$\mathbf{S}_i = \mathrm{softmax}(\frac{\mathbf{Q}_i \mathbf{K}_i^{\mathrm{T}}}{\sqrt{d_k}})\mathbf{V}_i \tag{8}$$

where $\mathbf{W}^{\mathrm{Q}}$, $\mathbf{W}^{\mathrm{K}}$, $\mathbf{W}^{\mathrm{V}} \in \mathbb{R}^{d_k \times d_h}$ are learnable matrices for linear projection and d_k is the dimension of key vectors. For different Self-attention modules in trajectory evaluation, a_i, b_i, and c_i denote the corresponding feature vectors of the given scene context entity. (More details can be found in https://github.com/Joe12138/PRISC-Net-V1)).

All possible trajectories $\mathcal{T}$ outputted by the generator are scored using a maximum entropy model. The interaction features $\mathcal{IF}$ extracted by the Self-Attention modules are concatenated and used as the input of the scoring model:

$$\xi(\tau|\mathcal{IF}) = \frac{\exp\{g(\mathcal{IF}, f(\tau))\}}{\sum_{\tau' \in \mathcal{T}} \exp\{g(\mathcal{IF}, f(\tau'))\}} \tag{9}$$

where $g(\cdot)$ is implemented using a 3-layer MLP [34], and the function $f(\cdot)$ is defined as follows:

$$f(\tau) = \frac{\exp(-\mathrm{D}(\tau, \mathrm{t_{GT}})/\sigma)}{\sum_{\tau' \in \mathcal{T}} \exp(-\mathrm{D}(\tau', \mathrm{t_{GT}})/\sigma)} \tag{10}$$

where σ is the temperature factor, and $D(\cdot)$ is the accumulated squared distance error between the predicted and ground-truth trajectories. The loss function for training the proposed overall trajectory evaluator is as follows:

$$\mathcal{L}_{\mathrm{E}} = \mathcal{L}_{\mathrm{CE}}(\xi(\tau|\mathcal{IF}), f(\tau)) \tag{11}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss, which measures the probabilistic deviation of the estimated score from the score labels. Given the evaluated scores, the predicted candidate trajectories are ranked in descending order, and the top $\mathcal{N}_{\mathcal{T}}$ trajectories are selected as the ultimate trajectory predictions of $\mathbf{v}_{\mathrm{tar}}$, along with their probabilities estimated using their scores.

VI. Experiments

A. Experiment Setup

1) Datasets and Testing vehicle : To evaluate the effectiveness and performance of the proposed approaches, the proposed PRISC-net is validated on both real-world vehicle motion datasets and simulated dataset. In addition, we have implemented the proposed trajectory prediction framework on a testing vehicle and conducted real-world road tests. Details of the datasets and testing vehicles are as follows:

- *Datasets:* We evaluated the proposed framework for predicting the future trajectories of motor vehicles using three datasets: two real-world datasets and an in-house simulated dataset.
  a) Real-world Datasets: In this work, the *INTERACTION* motion prediction dataset [19] and exits and entries drone (*exiD*) dataset [44] are used for the evaluation. Concretely, INTERACTION dataset [19] contains the labeled trajectories of traffic agents and HD map information in highly interactive real-world scenarios, including roundabout, signalized/unsignalized intersections as well as highway/urban merging and lane change, recorded at various locations of the different countries. All training and test trajectory data in INTERACTION are provided in the form of 4-second state sequences sampled at 10 Hz. Similarly, the exiD dataset collects trajectories of traffic agents in on- and off-ramp scenarios in German Autonahn [44]. All training and test trajectory data in the exiD dataset are sampled at 25 Hz [44].
  b) Simulation Dataset: We recorded the motion state of 100 vehicles and all traffic signals in a simulated scenario from the Intel CARLA simulator [21]. All training and test trajectory data are provided in the form of 300-second and 100-second state sequences sampled at 10 Hz, respectively. Therefore, there are 296,000 and 96,000 cases in the training and test dataset, respectively.
- *Tesing vehicle:* The configuration of the testing vehicle used for road test is shown in Fig 5. The testing vehicle is equipped with an onboard 128-channel LiDAR, four milliwave radars, a GNSS-inertial navigation module and an onboard computer. During the road test, the relative positions of surrounding vehicles are extracted using the 3D point clouds captured by the LiDAR, and these positions in the LiDAR coordinate system are then transformed into x-y coordinates in the global reference frame, using the absolute ego-vehicle location provided by the GNSS-inertial system. The velocities of surrounding vehicles are measured by the milliwave radars. The proposed PRISC-Net trajectory prediction algorithms are programmed in Python and C++ and implemented in the

Fig. 5. Configuration of the testing vehicle for the real-world road test, equipped with a 128-channel LiDAR, a high-precision GNSS-inertial navigation system, and an onboard computer for hosting the software.

onboard computer with an Intel i7-6700 processor and 32 GB of memory.

2) Evaluation Metrics: We evaluate the proposed approach based on two types of metrics: the prediction accuracy metrics, including miss rate (MR) [4], minimum average and final displacement error (minA/FDE) [4], and the feasibility metrics: the traffic rule violation rate (TRV). The evaluation metrics are defined as follows:

- *minimum Average Displacement Error* (minADE): the l_2 distance between the most possible trajectory among k predicted trajectories and the ground truth, averaged over all future time steps ($k = 6$ in this paper).
- *minimum Final Displacement Error* (minFDE): the l_2 distance between the most possible trajectory among k trajectories and the ground-truth at the final time step of prediction ($k = 6$ in this paper).
- *Miss Rate* (MR): the ratio of cases where the displacement between the predicted endpoint and the ground-truth endpoint exceeds the pre-defined threshold β ($\beta = 2.0m$ in this paper).
- *Traffic Rule Violation Rate* (TRV): the ratio of scenarios where any predicted trajectory violates traffic rule or scene context constraints. Typical cases include entering non-drivable areas, speeding, and retrograding. *Entering non-drivable area* is the case that any point of any predicted trajectory lies in the non-drivable area. *Speeding* means that the speed of any point in any predicted trajectory exceeds the speed limit. *Retrograding* represents cases in which the predicted trajectories drive against the direction of traffic. In this work, a predicted trajectory is considered retrograding if the angle between the driving direction of any point of that trajectory and the lane reference exceeds 90 degrees.

In addition, the most possible trajectory is defined as the one that has the minimum final displacement error (FDE).

3) Data Format: The input, output, and intermediate data are shown in Fig. 2. Formats of these data are as follows:

a) Map Data: The proposed PRISC-Net is comptatible with HD maps in the Lanelet2 [45] or OpenDRIVE vector map formats[3]. The map data encodes roads (long) using a representation of lanelets (short), and the connectivity among lanelets is also defined. The HD map also encodes implicit scene context (traffic signs, driving directions, and speed limits).

b) State data of Vehicles: The state data of vehicles consist of timestamped 2-D position (x, y), heading (θ) and velocity (v). The state sequences are sampled at a frequency of 10 Hz (for INTERACTION dataset [19], simulation dataset, and real-road test) or 25 Hz (for the exiD dataset [44]).

4) Implementation Details: All learning-based models are trained on an NVIDIA TITAN V100 GPU with 12 GB memory, and the implementation details for each stage are as follows:

a) Candidate Path Target Predictor: For candidate target sampling, two points are sampled every meter from lane centerlines. The number of hidden units is set to 64 for all 3-layer MLPs. The overall target predictor is trained for 80 epochs using Adam [46] optimizer with the batch size and initial learning rate set to 128 and 1×10^{-3}, respectively.

b) Trajectory Generator: In our experiment, the coefficients in Eq. 4, 5, 6 are set as: $k_j = 0.1$, $k_v = k_s = 1$, $\alpha_1 = 5$, $\alpha_2 = 2$ and $\alpha_3 = \frac{\pi}{6}$.

c) Trajectory Evaluator: The INTERACTION dataset provides an observed state sequence with a time interval of $\triangle T = 0.1s$, and the continuous trajectories are discretized with the same time interval. All reachable path inputs are discretized with a distance interval of $\triangle D = 2m$. We train the evaluator for 80 epochs with a batch size of 128 and initial learning rate of 1×10^{-3}. The evaluator is optimized with Adam [46] with a decay of 10 every 10 epoch.

B. Comparison with State-of-the-art Methods

We compare the performance of the proposed PRISC-Net against three representative state-of-the-art interactive predictors: the PRIME [17], the DenseTNT [12], and the HEAT-I-R [32]. The PRIME utilizes a pipeline similar to our proposed PRSIC-Net, consisting of a model-based planner and a learning-based evaluator. It predicts multi-modal trajectories by jointly considering motion constraints, lane connectivity, and inter-agent interactions. The DenseTNT has achieved top ranks on several behavior forecasting benchmarks and won the 1^{st} place winner of the 2021 Waymo Motion Prediction Challenge. The HEAT-I-R is an end-to-end approach that utilizes inter-agent interaction, map, and vehicle state information to make trajectory predictions. A comparison of the performance of the trajectory predictors is shown in Table I.

1) Overview of evaluation: The evaluation results are summarized in Table I. These results indicate that our proposed PRISC-Net outperforms state-of-the-art methods in both prediction accuracy and feasibility metrics. In terms of minFDE and MR, PRISC-Net achieves state-of-the-art performance on both real-world and simulated datasets, compared with the other three methods. For minADE and TRV, the proposed

[3]More details are available at https://www.asam.net/standards/detail/opendrive/

TABLE I
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART APPROACHES ON REAL-WORLD AND SIMULATED TRAJECTORY DATASET.

Methods	k[1]	MM[1]	minADE↓[2]	minFDE↓[2]	MR↓[3]	TRV↓[3] (%)
Evaluated on the INTERACTION real-world test dataset						
PRIME [17]	6	✓	0.676	1.096	0.141	92.67
DenseTNT [12]	6	✓	0.322	0.897	0.091	22.65
HEAT-I-R [32]	1	✗	0.216	0.780	0.079	1.20
PRISC-Net (**ours**)	1	✓	0.273	0.743	0.058	**0.33**
	6		**0.214**	**0.425**	**0.029**	0.39
Evaluated on the exiD real-world test Dataset						
PRIME [17]	6	✓	13.885	19.178	0.956	**0.06**
DenseTNT [12]	6	✓	**0.709**	2.636	0.199	17.89
HEAT-I-R [32]	1	✗	7.922	15.703	0.903	100.00
PRISC-Net (**ours**)	1	✓	1.284	1.035	0.150	33.54
	6		1.272	**0.673**	**0.044**	34.23
Evaluated on the simulated test dataset						
PRIME [17]	6	✓	1.599	3.424	0.224	17.16
DenseTNT [12]	6	✓	1.882	3.307	0.286	76.24
HEAT-I-R [32]	1	✗	2.039	5.44	0.399	22.38
PRISC-Net (**ours**)	1	✓	1.429	3.849	0.277	**6.81**
	6		**1.200**	**3.084**	**0.223**	8.70

[1] k: number of predicted trajectories; MM: multi-modal prediction.
[2] minA/FDE: minimum average/final displacement error.
[3] MR: miss rate; TRV: traffic rule violation rate.

PRISC-Net outperforms the other three state-of-the-art methods on the INTERACTION and simulated dataset.

a) PRISC-Net vs. PRIME: In terms of prediction accuracy, the proposed PRISC-Net outperforms PRIME on all three datasets. Specifically, PRISC-Net achieves an average improvement of 61.38%, 55.88%, and 58.43% in minADE, minFDE, and MR on three test datasets. For feasibility metrics TRV, the proposed PRISC-Net outperforms PRIME on INTERACTION and simulated dataset, with an improvement of 99.58% and 49.30%.

b) PRISC-Net vs. DenseTNT: The proposed PRISC-Net outperforms DenseTNT in both prediction accuracy and feasibility metrics on the INTERACTION and simulated datasets. On exiD dataset, our proposed PRISC-Net outperforms DenseTNT in terms of minFDE and MR. Compared against DenseTNT, PRISC-Net achieves an average improvement of 44.61% and 56.37% in terms of minFDE and MR among all three datasets, and an average improvement of 34.89% and 93.43%in minADE and TRV on INTERACTION and simulated dataset, respectively.

c) PRISC-Net vs. HEAT-I-R: For fairness considerations, we have also conducted single-modal trajectory prediction experiments, and the results are summarized in Table I. Again, the proposed PRISC-Net outperforms HEAT-I-R in terms of minFDE, MR, and TRV on all three test datasets. Specifically, the proposed PRISC-Net achieves an average improvement of 42.47%, 46.85%, and 69.51% in terms of minFDE, MR, and TRV on all three test datasets, respectively.

2) Analysis of Experiment Results:

a) Understanding the comparison results with PRIME: As shown in Table I, the proposed PRISC-Net outperforms the baseline PRIME in all accuracy metrics on the three test datasets, with a decrease of minADE, minFDE and MR upt o 50% on real-world datatsets. It is also worth mentioning that our proposed PRISC-Net outperforms PRIME significantly in terms of scene context compliance (i.e., TRV) on the INTERACTION and simulated datasets.

To further investigate the strength of our proposed approach, we record the number of traffic violations of each type (shown in Table II) on the INTERACTION dataset. Moreover, qualitative prediction results of PRIME and the proposed PRISC-Net are plotted in Fig. 7 (*INTERACTION datatset*) and Fig. 11 (*simulated dataset*).

From the experiment results, we can conclude that:

- Our proposed PRISC-Net consistently improves the accuracy of trajectory prediction, as our path search mechanism provides better predicted reachable paths. When sampling candidate path targets in the proposed PRISC-Net, using both explicit and implicit scene context improves the precision of the reachable paths, making the predictor more scene context-compliant than the road geometry-dependent-only PRIME predictor. (A more detailed analysis is presented in Section VI-C1).
- The planning-based trajectory generator of our proposed PRISC-Net improves the quality of candidate trajectories due to the advantage of scene context-aware, optimization-based planning. First, unlike PRIME, which uses fixed parameters for trajectory generation, using dynamical parameters (e.g., scene-related final position of target agents) in the proposed PRISC-Net can improve the quality of generated candidate trajectories. Second, the optimization-based planner in the proposed PRISC-Net does not require precise reference lines and vehicle state heuristics, making it more robust than the curvature- and state-sensitive PRIME (Fig. 6). Third, fully utilizing scene context information helps the proposed PRISC-Net to predict more accurate final position of future trajectories. Therefore, experiment results indicate that our proposed planning-based trajectory generator provides

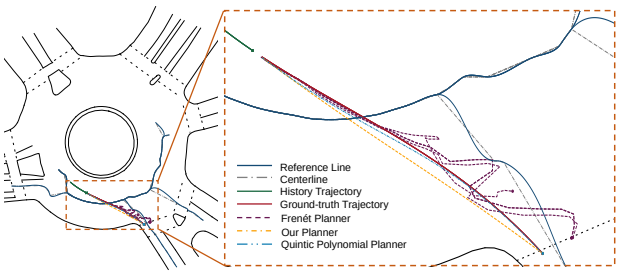


Fig. 6. Comparison of candidate trajectories generated by Frenét planner used in the baseline PRIME, quintic polynomial planner, and our proposed planner in PRISC-Net. (Tested on INTERACTION dataset [19]). Using our heuristic-free, scene context-enhanced optimization-based trajectory generator, the proposed PRISC-Net can generate more accurate and scene context-compliant candidate trajectories.

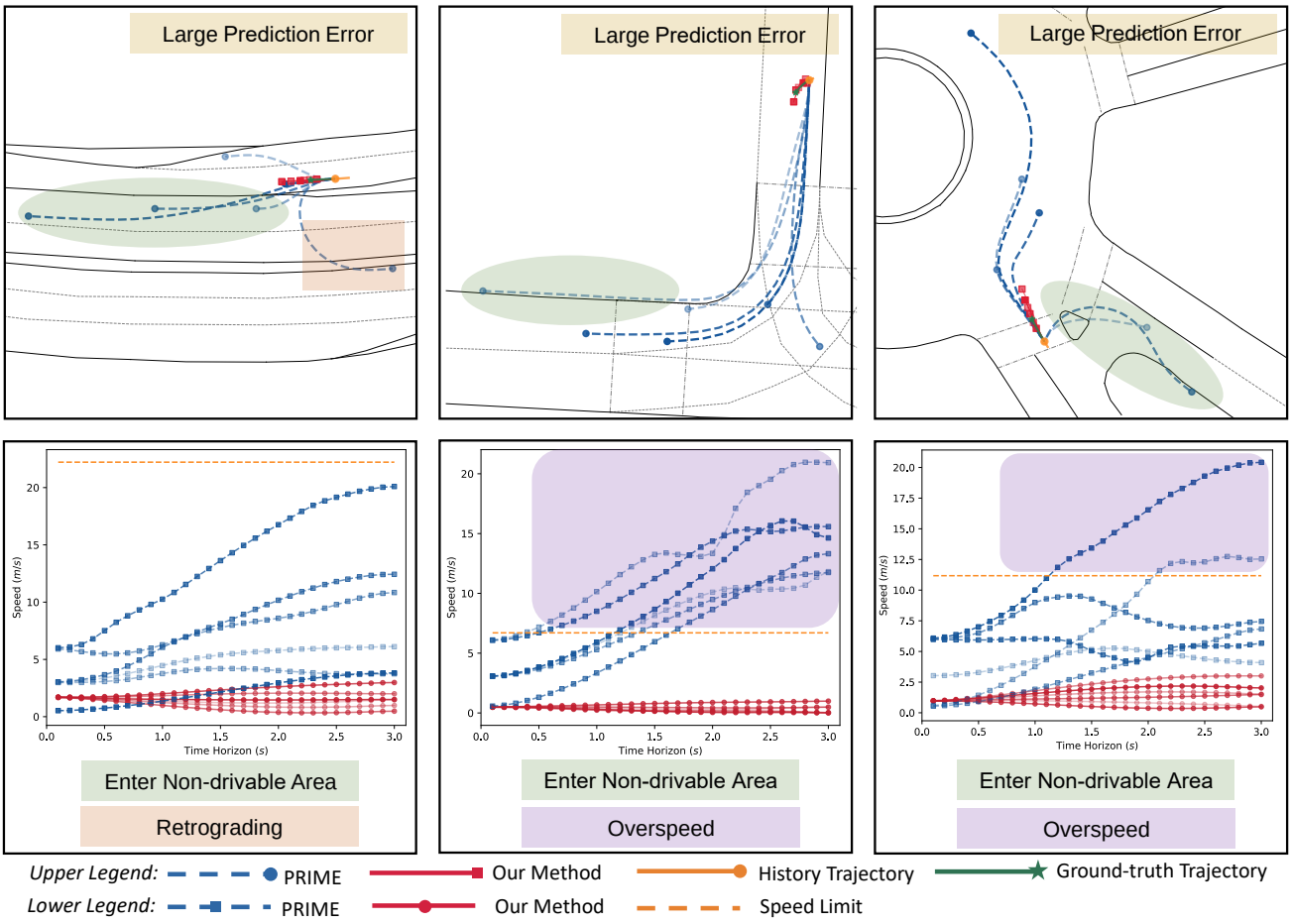


Fig. 7. Qualitative trajectory prediction results in the planar (*upper*) and speed dimension (*lower*) of the proposed PRISC-Net and baseline PRIME in merging (*left*), intersection (*middle*) and roundabout (*right*) scenarios. Green ellipses represent the cases where predicted trajectories (blue dash lines) enter the non-drivable areas. Orange rectangular and purple rounded rectangles represent cases where predicted trajectories perform retrograde motions and overspeed, respectively.

more accurate and scene context-compliant candidate trajectories, which helps the proposed PRISC-Net make more accurate predictions.

TABLE II
NUMBER OF TRAFFIC RULE VIOLATION CASES OF PREDICTED TRAJECTORIES ON THE INTERACTION DATASET

Methods	Entering ND Area* ↓	Speeding↓	Retrograding↓
PRIME	43,527	37,369	1,950
DenseTNT	2,588	15,051	2,123
HEAT-I-R	906	47	94
PRISC-Net (ours)	**286**	**32**	**26**

*ND: Non-drivable

b) Understanding the comparison results with DenseTNT: Table I indicates that our proposed PRISC-Net outperforms the baseline DenseTNT [12] with a large decrease of prediction error and traffic rule violations on both real-world INTERACTION and simulated datasets. On the exiD dataset, the proposed PRISC-Net also achieves better prediction accuracy (minFDE and MR), compared with the baseline DenseTNT.

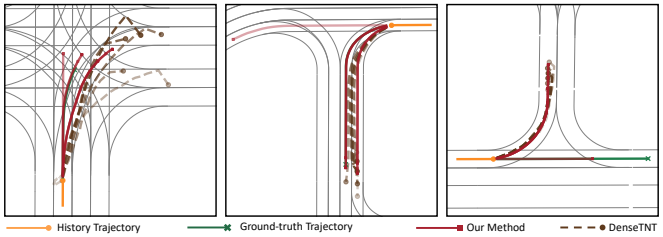


Fig. 8. Qualitative trajectory prediction results of the proposed PRISC-Net and the baseline DenseTNT in interactive scenarios. The predicted trajectories of our proposed method (*red square*) coincide better with the ground truth (*green cross*).

The qualitative trajectory prediction results of the baseline DenseTNT and the proposed PRISC-Net are plotted in Fig. 9 (*INTERACTION datatset*) and Fig. 11 (*the simulatied dataset*). From the qualitative (cf. Fig. 9 and Fig. 11) and quantitative results in Table I and Table II, we can conclude that:

- In the path target prediction stage, our proposed scene context-aware path search approach predicates more accurate path targets, which aids the proposed PRISC-Net in achieving better overall performance. In contrast, the baseline DenseTNT samples a larger number of path

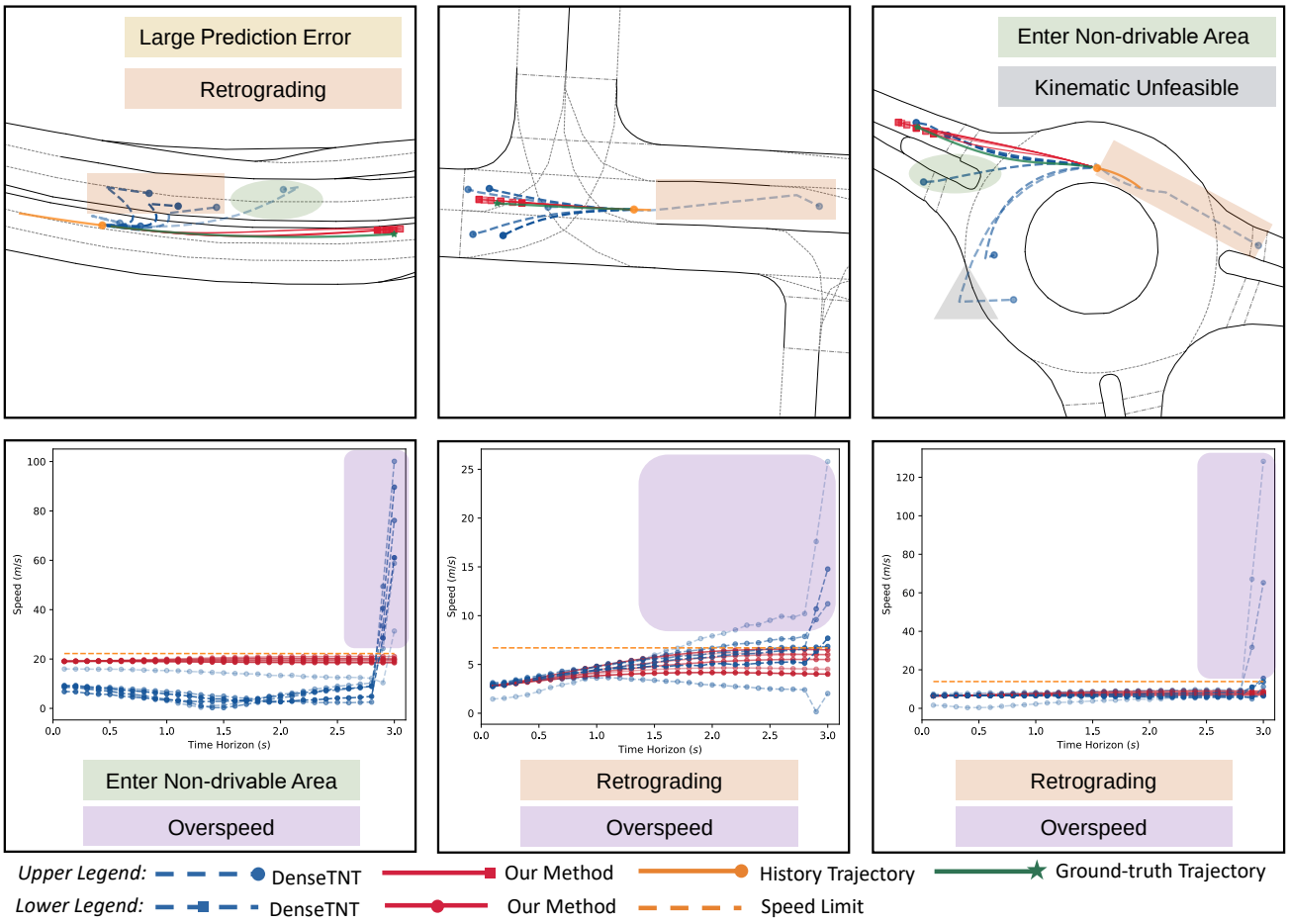


Fig. 9. Qualitative trajectory prediction results in the planar (*upper*) and speed dimension (*lower*) of the proposed PRISC-Net and baseline DenseTNT in merging (*left*), intersection (*middle*) and roundabout (*right*) scenarios. The green ellipse and gray triangle indicate the case where predicted trajectories enter non-drivable areas and are kinematically unfeasible, respectively. Orange rectangular and purple rounded rectangles represent the cases where predicted trajectories perform retrograding motions and overspeed, respectively.

targets. In fact, dense sampling typically degrades the overall performance, which is illustrated in Section VI-C1 and Fig. 13. In addition, the baseline DenseTNT samples candidate path targets from the current position along the entire possible path to the very end, without considering whether certain parts of the path are infeasible. In contrast, using the implicit scene context information, our proposed scene context-aware path search strategy avoids the dense sampling process and only searches the feasible part of paths constrained by scene context.

- Unlike the baseline DenseTNT that regresses trajectories in an end-to-end manner, our proposed PRISC-Net takes into account implicit scene context and kinematic constraints when generating candidate trajectories. Therefore, it can guarantee the kinematic and context feasibility of predicted trajectories (cf. Section V-B).
- Taking advantage of the interaction-aware trajectory evaluator, our proposed PRISC-Net can effectively capture the interaction among future trajectories. Therefore, the proposed PRISC-Net can generate more accurate multi-modal trajectories in highly interactive complex scenarios (Fig. 8).

c) Understanding the comparison results with HEAT-I-R: On the three test datasets, the proposed PRISC-Net consistently outperforms the baseline HEAT-I-R in terms of all evaluation metrics when predicting multiple trajectories (Table I).

From the qualitative study results shown in Fig. 10, 11 and statistics in Table II, we can conclude that:

- Fully utilizing both explicit and implicit scene context improves the prediction accuracy and feasibility. For explicit scene context, in contrast to the baseline HEAT-I-R that utilizes a rasterized map, the proposed PRISC-Net encodes the HD map as vectorized representation, which provides better structural features [12]. In addition, using the implicit scene context information, the proposed PRISC-Net can capture extra context features than the explicit map-dependent HEAT-I-R.
- Taking advantage of the joint model- and learning-based pipeline, the proposed PRISC-Net outperforms the pure end-to-end approaches. During the model-based path target prediction and candidate trajectory generation stage of our proposed framework, implicit scene context and kinematic constraints are effectively incorporated, making the predicted trajectories more accurate, feasible, and

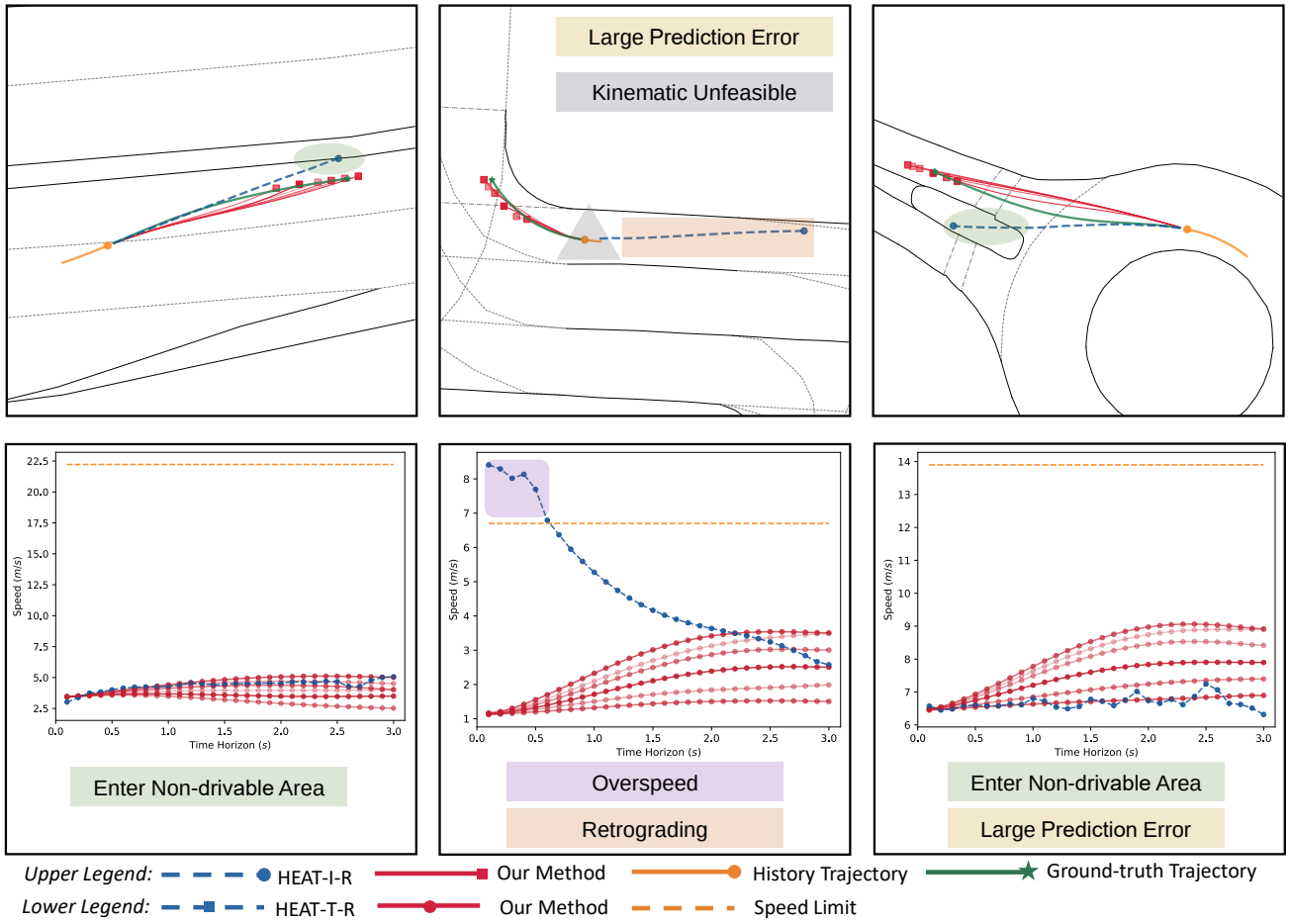


Fig. 10. Qualitative trajectory prediction results in the planar (*upper*) and speed dimension (*lower*) of the proposed PRISC-Net and baseline HEAT-I-R in merging (*left*), intersection (*middle*) and roundabout (*right*) scenarios. Green ellipses and gray triangles represent the cases where predicted trajectories (blue dash lines) enter non-drivable areas and are kinematically unfeasible, respectively. Orange rectangular and purple rounded rectangles represent the cases where predicted trajectories perform retrograding motions and overspeed, respectively.

context-compliant.

C. Ablation Study

To evaluate the effectiveness and strength of each component of the proposed PRISC-Net, we conduct an ablation study and compare the proposed components (reachable path search, candidate target prediction, feasible trajectory generation, and trajectory evaluation) with those in the baseline PRIME [17] on the INTERACTION dataset.

1) Reachable Path Search: To improve the efficiency of path target prediction and guarantee the scene-context compatibility of candidate paths, the proposed PRISC-Net utilizes implicit scene context (e.g., speed limits, traffic signs) to restrict the search of reachable paths. To validate its effectiveness, we compare the performance of the candidate path target predictors with reachable paths searched by the proposed PRISC-Net and the baseline PRIME (in Fig.13). The target prediction with reachable paths searched by PRISC-Net outperforms the baseline PRIME in accuracy. In addition, the training of the target predictor of the proposed PRISC-Net is more time-efficient. Compared to the baseline PRIME which takes about 30 minutes for each iteration during training, the training time of the target predictor of PRISC-Net is approximately 6 minutes per iteration. To investigate the factors affecting the training efficiency, we compared the length of predicted paths and the number of candidate targets sampled by PRIME and PRISC-Net:

- *Path length:* According to Table III, the average length of reachable path searched by PRISC-Net is approximately 30 meters shorter than those by PRIME (Fig.12), greatly improving efficiency. This conclusion works for each type of scene in INTERACTION. The average length of reachable paths searched by PRIME and PRISC-Net in the roundabout, intersection, merging scene are 121.7, 89.3, 96.5 m and 52.8, 77.1, 80.1 m, respectively (tested on INTERACTION validation set). It is worth mentioning that for roundabout scenarios, the length of reachable paths in PRISC-Net is only half of the length in baseline PRIME. Since roundabout scenarios are more interactive, they involve lower speed limits and denser traffic signs. Such factors must be considered for a more efficient search of candidate paths.
- *Number of candidate path targets:* According to Table III, the number of sampled candidate targets of PRISC-Net is only one-third of the number of targets of the

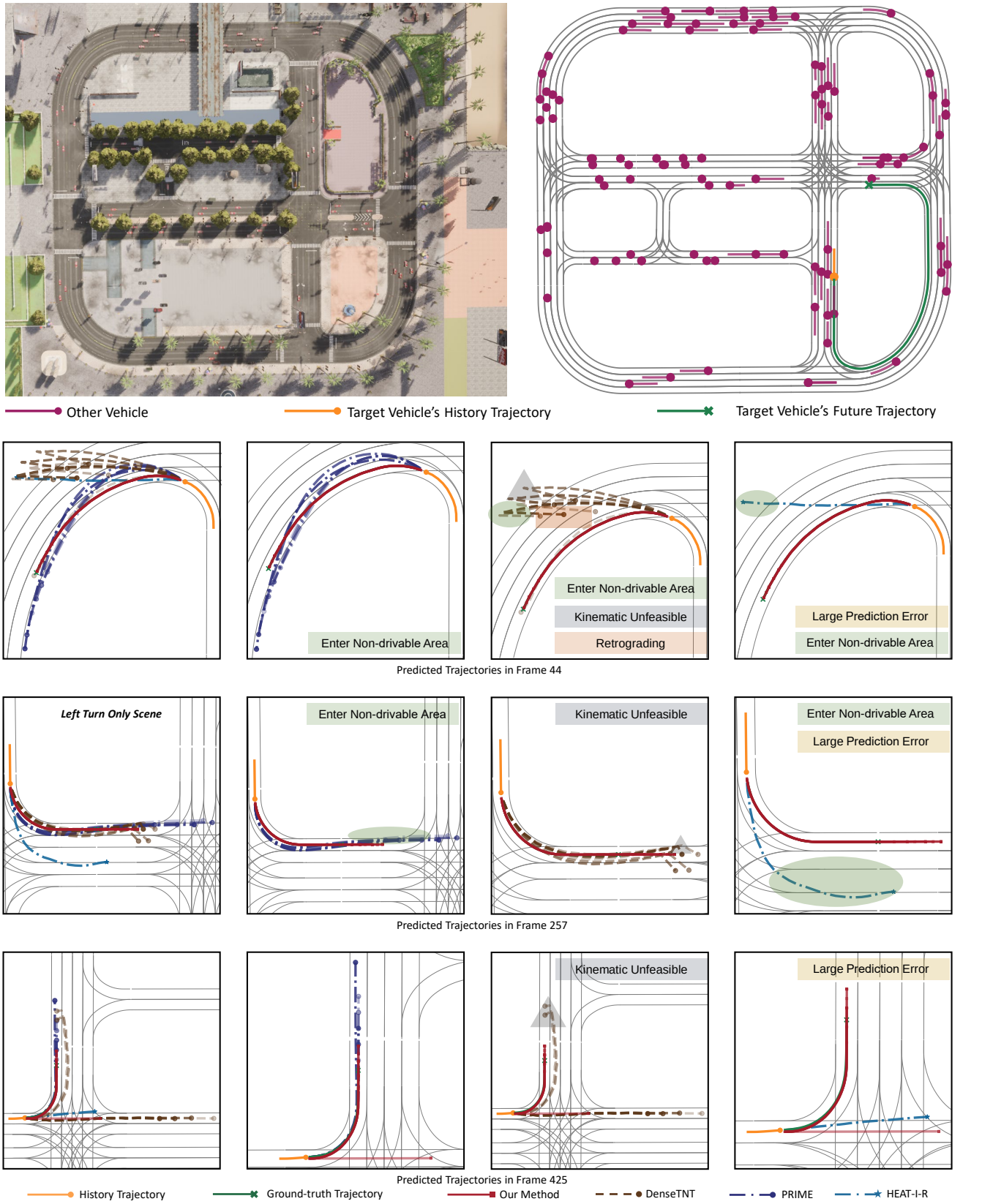


Fig. 11. Overview of the map and recorded dataset in a simulated scenario (*first row*) and qualitative prediction results at frame 44 (*second row*), 257 (*third row*) and 425 (*forth row*). For qualitative prediction results, the first figure compares trajectories generated by all methods; the second, third, and fourth figures show comparisons between our proposed method and the baseline PRIME, DenseTNT, and HEAT-I-R. The green ellipse and gray triangle indicate the case in which predicted trajectories enter non-drivable areas and are kinematically unfeasible, respectively. The orange rectangles represent the cases in which predicted trajectories perform retrograde motions.

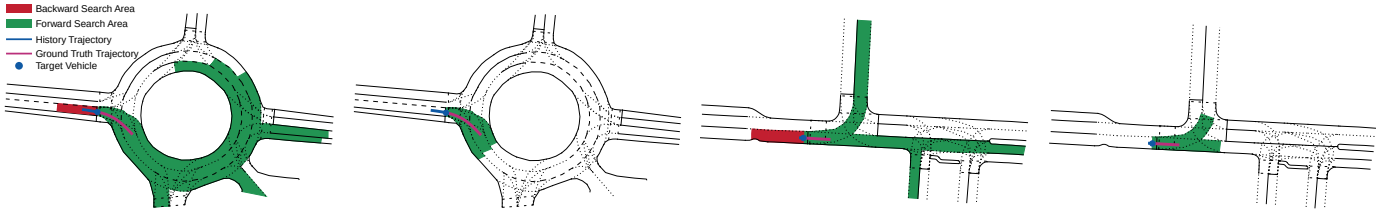


Fig. 12. Qualitative reachable paths searched by the baseline PRIME (*1 and 3*) and our proposed PRISC-Net (*2 and 4*) in roundabout scene (*1 and 2*) and intersection scene (*3 and 4*).

TABLE III
AVERAGE LENGTH AND NUMBER OF CANDIDATE PATH TARGETS SAMPLED FROM PATHS SEARCHED BY BASELINE PRIME AND PRISC-NET

Path Type	Dataset	Average Length	Number of Candidate Path Targets		
		AVG±STD[1]	AVG±STD[1]	MAX[2]	MIN[2]
PRIME	Train	98.99±34.13	1,017.61±503.57	8,561	11
Path	Val	99.03±34.73	959.97±481.77	8,610	31
PRISC-Net	Train	71.65±25.52	322.59±260.54	1,601	5
Path	Val	70.76±24.99	311.78±251.43	1,490	6

[1] AVG: average value; STD: standard deviation.
[2] MAX/MIN: maximum/minimum value.

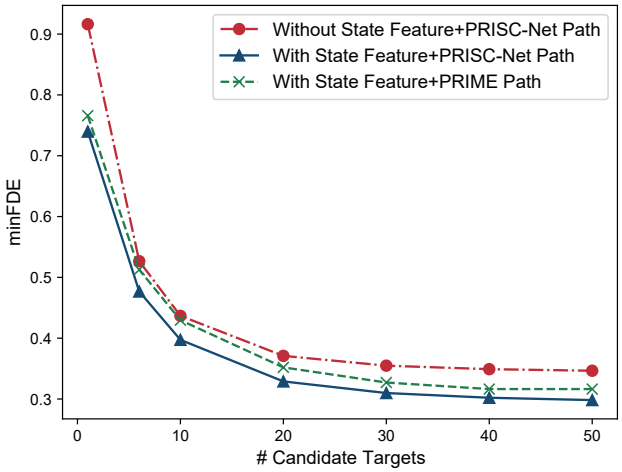


Fig. 13. The performance of candidate target predictor with different inputs.

baseline PRIME in both training and test sets. This is also an important factor that improves the efficiency of the proposed approach.

2) Candidate Path Target Prediction: As described in Section V-A2, the proposed PRISC-Net extracts rich scene features with implicit context states. The performance comparison of candidate target prediction of approaches with and without state features is shown in Fig.13. These results indicate that by incorporating additional state features, the proposed PRISC-Net achieves better accuracy in target prediction.

3) Feasible Trajectory Generation: We compare the trajectories generated by the baseline PRIME (based on basic Frenét planner [38] with fixed parameters for each traffic scenario), and our proposed PRISC-Net with an optimization-based planner and dynamic parameters. As shown in Table IV,

TABLE IV
THE QUALITY OF PREDICTED TRAJECTORIES GENERATED BY PRIME AND PRISC-NET

Method	Training Set		Validation Set	
	minADE↓	minFDE↓	minADE↓	minFDE↓
baseline PRIME	0.719	1.118	0.716	0.819
PRISC-Net (ours)	**0.205**	**0.322**	**0.194**	**0.321**

the trajectories generated by our PRISC-Net achieve better accuracy than those of baseline PRIME, on both INTERACTION training and validation set. Therefore, using dynamic parameters and scene-context information in the optimization-based planner greatly improves the accuracy of the predicted candidate trajectories, even in the presence of inaccurate reference lines and vehicle state heuristics (Similar results can also be found in Fig. 6).

We conducted further comparative experiments to validate these factors. In these experiments, ground-truth parameters are given to these two planners to generate trajectories. The experiment results are as follows: the average minADE of our planner and Frenét planner [38] are 0.19±0.23 and 1.57±27.83, respectively. In addition, the average minFDE are 0.00±0.00 and 3.52±52.72, respectively (Numbers after ± are standard deviation). These results further indicate that the proposed planner-based PRISC-Net can generate feasible trajectories with higher accuracy, without requiring precise trajectory parameters.

4) Trajectory Evaluation: To validate the effectiveness of the trajectory evaluator employed by the proposed PRISC-Net in modeling agent-scene interactions, we compare the performance of learning-based trajectory evaluators with and without features $\mathcal{F}_{\text{sce}}$ outputted by the scene feature extractor. As shown in Table V, the trajectory evaluator without $\mathcal{F}_{\text{sce}}$ outperforms feature-based evaluators with lower minFDE and MR, and achieves comparative minADE. The results indicate that the trajectory evaluator employed in the proposed PRISC-Net can effectively capture agent-agent and agent-scene interactions without requiring additional feature extraction.

D. Real-world Road Test

To demonstrate the effectiveness of our proposed PRISC-Net in real-world autonomous driving applications, we conducted real-world road tests based on the testing vehicle shown in Figure 5 and Section VI-A.

TABLE V
COMPARISON THE PERFORMANCE OF TRAJECTORY EVALUATOR WITH AND WITHOUT LEARNED SCENE FEATURES

Module	minADE[1]↓	minFDE[1]↓	MR[2]↓
With $\mathcal{F}_{\text{sce}}$	**0.208**	0.526	0.036
Without $\mathcal{F}_{\text{sce}}$	0.214	**0.425**	**0.029**

[1] minA/FDE: minimum average/final displacement error.
[2] MR: miss rate.

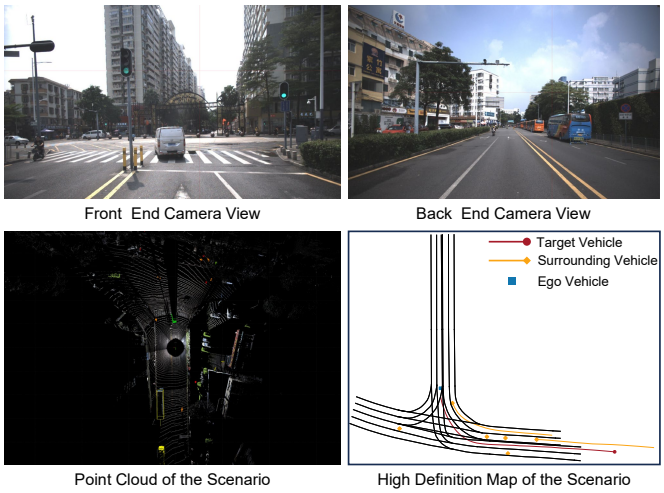


Fig. 14. Sensing data of the road test scenario captured by the testing vehicle sensors: RGB images (*upper left* and *upper right*), LiDAR point clouds (*lower left*). High-definition map (*lower right*) is also utilized.

1) Testing Scenario: The road test scenario is shown in Fig. 14. It contains an intersection with multiple directional traffic controlled by traffic signals. We select the van driving in front of the ego vehicle as the target vehicle (shown in the front camera view of Fig. 14) which makes a left turn at the intersection. The target vehicle's future motion in the next 20 seconds (the lower right subfigure of Fig. 14) is continuously predicted by the proposed PRISC-Net, which runs on the vehicle-mounted computer.

2) Experiment Results: The quantitative and qualitative results of real-world road test are shown in Table VI and Fig. 15, respectively. For both single-modal and multi-modal predictions in the 20-second continuous prediction cycle, the proposed PRISC-Net achieves minADE and minFDE of less than 0.4m and 0.1m, respectively, with approximate zero MR and TRV. The test results indicate that the proposed PRISC-Net is effective in predicting accurate and feasible motions of surrounding vehicles in real-world applications.

3) Runtime Analysis: During operation, the inference time of the proposed PRISC-Net is affected by the complexity of the scenario, the density of path target sampling, and candidate trajectory generation density. It takes 588 ms (on average) to predict a target vehicle's future motion, by running the codes in Python with a single thread on the vehicle-mounted computer (Section VI-A1). More specifically, the average time cost of target prediction, trajectory generation and evaluation is 288, 99, and 201 ms, respectively. Therefore, the proposed PRISC-Net achieves desirable computational efficiency and shows good potential for real-time autonomous driving applications. In addition, PRISC-Net could be implemented with a more efficient programming language (e.g., *C++*) with a parallel computing mechanism to reduce the time cost further.

TABLE VI
PERFORMANCE OF PRISC-NET ON REAL-WORLD ROAD TEST

Method	k	minADE	minFDE	MR	TRV(%)
PRISC-Net	1	0.382	0.040	0.000	0.00
	6	0.380	0.032	0.000	0.00

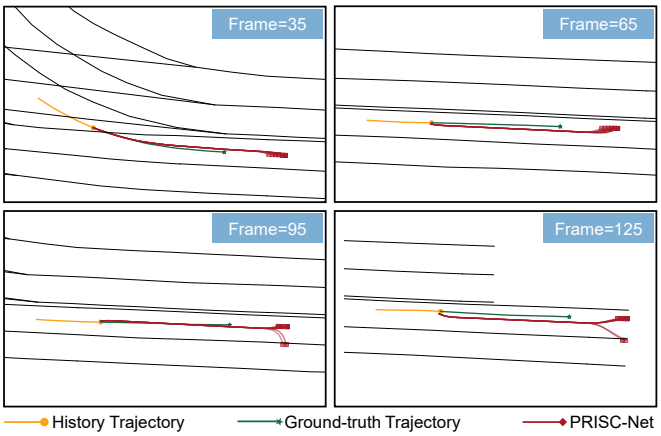


Fig. 15. Qualitative trajectory prediction results from real-world road tests at different time intervals.

E. Discussions

From the evaluation results, we have the following observations:

- *Fully encoding the scene context information can improve prediction accuracy.* The proposed PRISC-Net utilizes the abundant implicit scene context information to restrict the drivable area of the target vehicle and extract scene features, which helps to make more accurate candidate path target predictions, ultimately resulting in more accurate predicted trajectories.
- *The optimization-based planner helps to guarantee the feasibility of predicted candidate trajectories:* The proposed Optimization-based planner allows the proposed PRISC-Net to effectively incorporate kinematic and scene-context constraints, making the generated candidate trajectories more robust and feasible.
- *Modeling complex interactions can greatly improve the quality of predicted trajectories.* The agent-to-agent and agent-to-environment interactions are effectively modeled by the attention mechanism in the proposed trajectory evaluator, which aids the proposed PRISC-Net in predicting more reasonable multi-modal trajectories in complex and interactive scenarios.

VII. CONCLUSION

This paper has presented a scene context-aware behavior prediction framework for forecasting surrounding vehi-

cles' future trajectories in highly interactive and complex scenarios. The proposed PRISC-Net combines the strength of both model- and learning-based approaches to generate kinematic feasible, context-compliant, and interaction-aware trajectory predictions. The proposed candidate path target predictor can fully utilize scene context to make accurate and context-compliant target waypoint predictions. The proposed trajectory generator can generate kinematic feasible candidate trajectories. Finally, the learning-based trajectory evaluator can capture complex interactions and generate accurate final predictions. We evaluated the proposed framework on real-world and simulated behavior datasets, and its effectiveness is also demonstrated in road test via implementations on a testing vehicle. Experimental results show that the proposed PRISC-Net outperforms the state-of-the-art end-to-end methods in terms of prediction accuracy, feasibility, and scene context compliance.

Taking advantage of the joint model- and learning-based pipeline and the scene context awareness, our proposed framework shows good potential for trajectory prediction in real-world autonomous driving applications in complex scenarios, and it is scalable for other applications beyond the autonomous driving domain.

For future work, one promising direction is to address the diversity of traffic participants by introducing a category-specific attribute encoder since the behavior patterns of different traffic participants vary broadly. Another possible work is incorporating an adaptation mechanism to handle corner cases where the target vehicles do not follow traffic rules.

REFERENCES

[1] Y. Tian, X. Li, H. Zhang, C. Zhao, B. Li, X. Wang, X. Wang, and F.-Y. Wang, "Vistagpt: Generative parallel transformers for vehicles with intelligent systems for transport automation," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 9, pp. 4198–4207, 2023.

[2] L. Chen, Y. Li, C. Huang, B. Li, Y. Xing, D. Tian, L. Li, Z. Hu, X. Na, Z. Li, S. Teng, C. Lv, J. Wang, D. Cao, N. Zheng, and F.-Y. Wang, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1046–1056, 2023.

[3] H. Ben-Younes, É. Zablocki, M. Chen, P. Pérez, and M. Cord, "Raising context awareness in motion forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4409–4418.

[4] 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*, 2022, pp. 8823–8833.

[5] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[6] J. Zhao, T. Qu, X. Gong, and H. Chen, "Interaction-aware personalized trajectory prediction for traffic participant based on interactive multiple model," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2184–2196, 2023.

[7] M. Koschi and M. Althoff, "Set-based prediction of traffic participants considering occlusions and traffic rules," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 249–265, 2021.

[8] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu, and L. Chen, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3692–3711, 2023.

[9] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018.

[10] 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 Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[11] 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.

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

[13] L. Fang, Q. Jiang, J. Shi, and B. Zhou, "Tpnet: Trajectory proposal network for motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6797–6806.

[14] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 074–14 083.

[15] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.

[16] D. Choi, T.-H. An, K. Ahn, and J. Choi, "Future trajectory prediction via rnn and maximum margin inverse reinforcement learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 125–130.

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

[18] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 239–10 251, 2022.

[19] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[20] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.

[21] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[22] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3461–3475, 2023.

[23] L. Guo, C. Shan, T. Shi, X. Li, and F.-Y. Wang, "A vectorized representation model for trajectory prediction of intelligent vehicles in challenging scenarios," *IEEE Transactions on Intelligent Vehicles*, pp. 1–6, 2023.

[24] X. Li, H. Duan, B. Liu, X. Wang, and F.-Y. Wang, "A novel framework to generate synthetic video for foreground detection in highway surveillance scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 5958–5970, 2023.

[25] X. Li, Y. Tian, P. Ye, H. Duan, and F.-Y. Wang, "A novel scenarios engineering methodology for foundation models in metaverse," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 4, pp. 2148–2159, 2023.

[26] R. Song, X. Li, X. Zhao, M. Liu, J. Zhou, and F.-Y. Wang, "Identifying critical test scenarios for lane keeping assistance system using analytic hierarchy process and hierarchical clustering," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2023.

[27] X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F.-Y. Wang, "From features engineering to scenarios engineering for trustworthy ai: I&i, c&c, and v&v," *IEEE Intelligent Systems*, vol. 37, no. 4, pp. 18–26, 2022.

[28] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *Conference on Robot Learning*. PMLR, 2020, pp. 86–99.

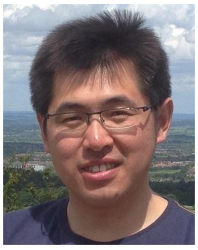
[29] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *Proceedings of*

the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2095–2104.
[30] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8454–8462.
[31] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Home: Heatmap output for future motion estimation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 500–507.
[32] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9554–9567, 2022.
[33] 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.
[34] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.
[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[36] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, 2021.
[37] Z. Li, Y. Wang, and Z. Zuo, "Interaction-aware prediction for cut-in trajectories with limited observable neighboring vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2148–2161, 2023.
[38] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 987–993.
[39] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.
[40] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.
[41] 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.
[42] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.
[43] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.
[44] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exid dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 958–964.
[45] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1672–1679.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Wenxing Lan received his B.Eng. degree from Southern University of Science and Technology (SUSTech), China in 2020. He is pursuing his Ph.D. at the Department of Computer Science and Engineering and Research Institute of Trustworthy Autonomous Systems of SUSTech. His research interests include evolutionary computation, vehicle detection, and trajectory prediction in autonomous driving.

Dachuan Li (Member, IEEE) received his Ph.D. in control science and engineering from Tsinghua University (2015), China. He is currently an assistant professor (research track) at the Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, China. He was a postdoctoral researcher at California PATH and Institute of Transportation Studies, University of California, Berkeley from 2016 to 2018. His research interests include autonomous driving vehicles, trustworthy autonomous systems, decision-making and motion planning.

Qi Hao (Member, IEEE) received the B.E. and M.E. degrees in electrical and computer engineering from Shanghai Jiao Tong University, Shanghai, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from Duke University, Durham, NC, USA, in 2006., He was a Post-Doctoral Trainee at the Center for Visualization and Virtual Environment, University of Kentucky, Lexington, KY, USA. He was an Assistant Professor with the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL, USA. He is currently a Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen, China. His research interests include intelligent sensing, machine learning, and autonomous systems.



Dezong Zhao (Senior member, IEEE) (M'12-SM'17) received the B.Eng. and M.S. degrees from Shandong University in 2003 and 2006, respectively, and the Ph.D. degree from Tsinghua University in 2010, all in Control Engineering. He was a Lecturer in Intelligent Systems with Loughborough University. He is currently a Reader in Autonomous Systems with the University of Glasgow. His research interests include connected and autonomous vehicles, robotics, machine learning and control engineering. He has been an EPSRC Innovation Fellow since 2018 and a Royal Society-Newton Advanced Fellow since 2020.

Bin Tian received the B.S. degree from Shandong University, Jinan, China, in 2009, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014. He is currently an Associate Professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include automated driving, vision sensing and perception, and machine learning.