Efficient Speed Planning for Autonomous Driving in Dynamic Environment With Interaction Point Model

Yingbing Chen, Ren Xin, Jie Cheng, Graduate Student Member, IEEE, Qingwen Zhang, Member, IEEE, Xiaodong Mei, Graduate Student Member, IEEE, Ming Liu, Senior Member, IEEE, and Lujia Wang, Member, IEEE

Abstract—Safely interacting with other traffic participants is one of the core requirements for autonomous driving, especially in intersections and occlusions. Most existing approaches are designed for particular scenarios and require significant human labor in parameter tuning to be applied to different situations. To solve this problem, we first propose a learning-based Interaction Point Model (IPM), which describes the interaction between agents with the protection time and interaction priority in a unified manner. We further integrate the proposed IPM into a novel planning framework, demonstrating its effectiveness and robustness through comprehensive simulations in highly dynamic environments.

Index Terms—Autonomous vehicle navigation, integrated planning and learning, motion and path planning.

I. INTRODUCTION

PLANNING safe trajectories for autonomous vehicles (AVs) in complex environments is a challenging task. As demonstrated in Fig. 1, it requires the planning system to have the ability to safely react in real-time under various scenarios, such as intersection, merging, and traversing areas with a limited field of view (FoV).

A typical strategy in solving these problems is to transform traffic rules and prediction results into constraints in an optimization problem [1], [2]. Although this method can generalize to different situations, it does not model the prediction and interaction uncertainties well. Another strategy applies the POMDP-based methods [3], [4], which searches for the best reaction trajectory by simulating the traffic interactions in the belief space.

However, these methods introduce a high computational burden, and are hard to reflect the interaction patterns using manually designed reward functions.

To better capture the interaction among traffic participants, it is essential to focus on modeling driving patterns around an “interaction point” without traffic regulations. The interaction points are observable situations where at least two road users intend to occupy the same space in the near future [5]. Some works implicitly model the planning problem on these points. For instance, car-following models [6], [7] and lane-merging approaches [8] use different values of time gap (\( \Delta t \) in Fig. 2(a)) to generate the proper reactive behavior for AVs. Prediction work [9] analyzes the right-of-way of collision points in the intersection. However, these methods are tailored for specific scenarios. It brings a heavy load on parameter tuning when applying multiple methods in different situations.

When reacting to other road users at different speed and angles in various scenarios, human drivers generally show different degrees of caution that can be indicated by the time gap. Therefore, we focus on the following two questions. Question 1: When two agents intend to pass the same interaction point in various scenarios, what is the minimum time gap between them? Question 2: What determines the priority of agents to pass the interaction point in Question 1?

To answer these questions, we propose a novel Interaction Point Model (IPM), where Question 1 is solved by statistical analysis in real driving records, and Question 2 is explained using a multilayer perceptron (MLP) network. Then, based on IPM, we present a unified planning framework\(^1\) to achieve safe and interactive motion planning, which guides the ego vehicle pass an interaction point only when it has a higher priority than other traffic participants.

A. Contributions

Overall, we propose a unified interactive planning framework for AVs. The major contributions are:

- A novel and general interaction solution called IPM, which uniformly formulates the interaction in terms of the protection time and priority.
- A unified planning framework for AV in complex dynamic environments, including an efficient s-t graph search to calculate the trajectory distribution under traffic regulations, and a priority determination module based on the IPM to select a safe trajectory.

\(^1\)IPM model files and demonstration videos are available on the project website: https://github.com/ChenYingbing/IPM-Planner/https://github.com/ChenYingbing/IPM-Planner.
IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 7, NO. 4, OCTOBER 2022

Fig. 1. The AV (ego) follows the path to the goal with considerations of interactions with other traffic agents (e.g., $V_i$) at different scenarios (different interaction angles and the blind zone).

Fig. 2. Illustrations of interactions in traffic. (a) Demonstration of the interaction angle $\Delta \theta$ and the time gap $\Delta t$, where the AV goes along the red path and $V_i$ moves along the blue one. (b) The AV cannot determine its maneuver until it reaches the observation point $p_o$.

- Evaluations in unsignalized urban driving simulations showing our method’s computational efficiency, robustness, and effectiveness.

II. RELATED WORK

A. Interaction Representation

Existing methods used different ways to represent traffic interactions in planning. One strategy was to represent prediction results and simulation outcomes [4], [10] in the spatio-temporal (s-t) graph. Then, the sampling methods [11], [12] and optimization methods [13] were used to extract the best-cost or lowest-risk trajectory. Another strategy applied reactive or interactive models [3] to generate action for AVs directly. Methods based on elaborated rules [6], [14], game theory [15], and control theory [16] were widely used in autonomous driving. Compared to these methods, our work applies both s-t graph representation and interactive model (IPM). This combination makes it avoid deterministic representation [11], [13], [17] and limited expression of interaction [6], [14], [15], [16]. In addition, our method is highly efficient as it does not require low-efficient simulation [4], [10] and accurate collision checking [11]. It also outperforms learning-based methods [18] on interpretability, as the IPM is simpler and more intuitive.

B. Interactive Motion Planning for AVs

In interactive planning, the motion of other traffic agents and the ego vehicle is modeled as interdependent. This interdependency makes it impracticable to separate the problems of prediction and planning [3]. There were several interactive planning approaches. For the discussed POMDPs [3], [4], [10], interactions were reflected by time-consuming simulations but not modeled directly. In recent years, learning-based approaches [19], [20] have been promising to learn trajectory generation and prediction in a unified architecture. However, machine learning technologies lacked traceability when an error occurred in practice. Another was game-theoretic planning [21], [22], aiming to compute the Nash equilibrium solutions for all players iteratively. The major shortage of these methods was their high computational cost when the number of players increased. In addition, the game rules were elaborately designed for specific situations, making them hard to generalize to different scenarios.

Besides these, branch MPC [23] and fail-safe planners [24], [25], [26] emphasized the worst-case performance of AVs. For branch MPC [23], it was an optimization method over risk measures to minimize the worst-case expectation, which did not model the traffic interactions well. Instead of accurate risk avoidance, fail-safe planning [25], [26] were reactive planners that guaranteed AV’s safe reactions to all possible prediction uncertainties. It achieved this by keeping a collision-free maneuver available at all times. Following this idea, our method shows a conservative maneuver only when the interaction priority is estimated as low, which prevents the AV from too conservative behaviors [26].

III. OVERVIEW

An overview of the proposed planning framework is presented in Fig. 3. In this system, the IPM has two functions: 1) determining whether the AV has a higher priority at an interaction point and 2) extracting the speed limit if the AV intends to pass an occluded area. The framework functions as follows: Given the inputs of the path and speed limits, the spatio-temporal (s-t) graph search first outputs the valid trajectory distribution. Then, the interaction priority determination decides a safe and efficient trajectory for the AV. Finally, the trajectory is smoothed to meet the kinodynamic feasibility and is delivered to the controller for execution.

IV. THE PROPOSED INTERACTION POINT MODEL

We extract pairs of interaction data from the INTERACTION dataset [27] to analyze the protection time and priority in the interaction points. Two examples are shown in Fig. 4, where car following is a special case of the line overlap [1]. It should be noted that the proposed IPM is also applicable in one-to-many interactions.
yielding for others. As a result, we define

\[
\begin{align*}
\Delta t^+ &= \max \Delta t, \forall \Delta t \in T \cap \mathbb{R}^+, \\
\Delta t^- &= \min \Delta t, \forall \Delta t \in T \cap \mathbb{R}^-
\end{align*}
\]

where \(\Delta t^+_p\) denotes the maximum time gap when overtaking, \(\Delta t^-_p\) is the minimum time gap when giving way, and \(T\) is the set of possible values of the time gap in the data set. Then, \(\Delta t_p\) is modeled as functions of \(\Delta \theta\) and \(\Delta v\) respectively.

\[
\begin{align*}
\Delta t^+_p(\Delta v) &= C_1 [\Delta v^4 \Delta v^3 \Delta v^2 \Delta v 1]^T, \\
\Delta t^-_p(\Delta \theta) &= C_2 [\Delta \theta^2 \Delta \theta 1]^T, \\
\Delta t^+_p(\Delta v) &= C_3 [\Delta v^4 \Delta v^3 \Delta v^2 \Delta v 1]^T, \\
\Delta t^-_p(\Delta \theta) &= C_4 [\Delta \theta^2 \Delta \theta 1]^T
\end{align*}
\]

where \(\Delta t_p(\Delta v)\) and \(\Delta t_p(\Delta \theta)\) are represented as fitting curves, and \(C(.)\) is the fitting coefficient. Given the values of \(\Delta v\) and \(\Delta \theta\) together, the boundaries become

\[
\begin{align*}
\{\Delta t^-_p &= \min (\Delta t^-_p(\Delta v), \Delta t^-_p(\Delta \theta)), \\
\Delta t^+_p &= \max (\Delta t^+_p(\Delta v), \Delta t^+_p(\Delta \theta)).
\end{align*}
\]

More illustrations will be given in Section VII-B.

D. Speed Constraint At Observation Point

Recalling Assumption 1 and 2, before all occluded vehicles are observed, the agent needs to prepare for the worst-case in the future, that is, to give way to any emergent vehicles. As the example shown in Fig. 2(b), before the AV reaches the observation point \(p_o\), the remaining time \(T\) to reach \(p_j\) should be greater than \(\Delta t_p\). This means the speed upper bound

\[
\bar{v} = \frac{d^2_e}{T} \leq \frac{d^2_e}{\Delta t^+_p}, d^2_e \geq \text{Dist}(p_o, p_j),
\]

where \(d^2_e\) is the distance between the AV and the \(p_j\), \(\text{Dist}()\) returns the Euler distance, and the upper bound meets its minimum \(\bar{v} = \text{Dist}(p_o, p_j)/\Delta t^+_p\). In this equation, \(\Delta t^+_p\) can be quickly estimated by setting \(\Delta v\) to zero, which means there is no need to predict agents’ speed in the occluded area.

E. Interaction Priority

The definition of the interaction priority is from Question 2. It is treated as a conditional distribution \(P_j(\Delta t < 0|f)\), where \(f\) is the feature vector. Before representing the input features, we introduce two metrics: the minimum arrival time \(\min (t^+_i)\) and the maximum arrival time \(\max (t^+_i)\). These two metrics are calculated by the constant acceleration model

\[
\begin{align*}
\min (t^+_i) &= \sqrt{\frac{v^2}{a_i} + 2 \max(a_i)d^2_i - v^2}, \\
\max (t^+_i) &= \frac{v - \sqrt{v^2 + 2 \cdot \max(a_i)d^2_i}}{\max(a_i)}, \text{ s.t. } a_i \geq -\frac{v^2}{2 \cdot d^2_i},
\end{align*}
\]
Algorithm 1: S-t Graph Search.

1: **Notation**: open list $\mathcal{O}$, layer index $l$ and $f$, path
2: points $\{s_1, s_2, \ldots, s_k\}$, node list $\mathcal{V}$, the maximum
3: layer $k$, constants $t_h$ and $c_v$
4: **Initialize**: $\mathcal{O} \leftarrow \{q^0\}$, $\mathcal{V} \leftarrow \emptyset$, $f \leftarrow 0$
5: **for** $n_{iter} \in \{0, \ldots, \max_{iter}\}$ **do**
6:   $q^i \equiv (s_i, v, t)^i \leftarrow \text{POPBESTNODE}(\mathcal{O})$
7:   $f \leftarrow \min(f, l)$, $\mathcal{V}_l \leftarrow \mathcal{V}_l \cup q^i$
8:   **if** not OUTOFBOUNDS($q^i, l, k, t_h, c_v$) **then**
9:     $a \in \{a_1, \ldots, a_n\}$
10:    $q^c \leftarrow \text{FORWARDEXPANSION}(q^i, a, s_i, s_{i+1})$
11:    $J = \text{CALCULATECOST}(q^i, q^c)$
12:    **if** LOCALTRUNCATION($q^c, J, R_v, R_t$) **then**
13:      $\mathcal{O} \leftarrow \mathcal{O} \cup q^c$
14:      **end if**
15: **end for**
16: **return** $\mathcal{V}_f$, PATHTO($q^*$)

where $v_{0i}$ denotes the initial speed of agent $V_i$, $a_i$ is the possible
average acceleration before reaching the interaction point, and $\max(a_i)$ and $\min(a_i)$ are the maximum and minimum values of
acceleration. When the AV and $V_i$ intend to pass $P_j$, two features
named the overtaking ability $M^-$ and giving way ability $M^+$ are
defined as

$$M^-_j = \min(t'_c) - \min(t'_i) + |\Delta t'_p|,$$

$$M^+_j = \max(t'_c) - \max(t'_i) - |\Delta t'_p|,$$

(8)

where the expression of $M^-$ indicates that the overtaking needs
to suffer the protection time $\Delta t', \text{and } M^+$ is the time difference
when the two agents slow down simultaneously. In this equation,
when $M^-$ is less than zero, the AV can safely occupy the
interaction point in advance because it can arrive much earlier
than $V_i$. When $M^-$ is positive, if the AV wants to take,
the communications mentioned in Assumption 1 and the traffic
regulated must be considered.

Our goal is to learn a binary classifier $P_j(\Delta t < 0|M^-, M^+)$. Since agents in the experiments are not as intelligent as in the
dataset, the classifier is modified as

$$\begin{cases} P_j(\Delta t < 0) = 0.0, M^- \geq t_m, \\ P_j(\Delta t < 0) = \text{MLP}(M^-, M^+), M^- < t_m. \end{cases}$$

(9)

where MLP is the trained 3-layer MLP model, $t_m = \min(C^{1}_r \cdot t_c + C^{2}_r, C^{3}_r)$ is the time threshold, and $C^{1}_r$, $C^{2}_r$, $C^{3}_r$ are constants ($C^{3}_r$ is positive). When the AV is far from the interaction point ($t'_c$ is large), it is assumed that the interaction follows patterns learned from the dataset ($M^-$ is less than $t_m$). In contrast, when $t'_c$ is small, it turns into a more conservative model in which $M^-$ plays a key role.

V. FORWARD SIMULATION

The minimum arrival time in the IPM uses a constant ac-
celeration model, which is not competent when the speed limit
changes. Hence, a spatio-temporal (s-t) graph search method is
applied to sample speed profiles along the path. This module
does not generate a collision-free speed profile but obtains the
future trajectory distribution under variational speed limits. As
shown in Fig. 5, we present the graph node as $q^i = (s_i, v, t)^i$, where
$v$ is the node index, $s_i$ is the accumulated distance, the subscript $l$
is the layer index, and $v$ and $t$ are the speed and timestamp values.

A. Path Point Selection

The first step is to select a sequence of path points $\{s_1, s_2, \ldots, s_k\}$ to build the search space, where $k$ is the max-
imum layer. This process considers the curvatures and locations
of the interaction points, which obeys: (i) The selected point should reflect the speed limit change between bends and
strait. (ii) The path point should involve the location of the
interaction point and observation point. (iii) The number of
layers should stay a suitable size. Too many layers will increase
the computer amount, while a too-small number will reduce the
search space.

B. Speed Constraints

During forward simulation, traffic elements, e.g., traffic lights,
stop signs, and speed limits, can be transformed into spatio-
temporal constraints in the search space [13]. This work involves
two kinds of velocity constraints: one comes from the curvature,
and the other from the observation point in (5). We denote $\varepsilon_{\text{max}}$
as the maximum lateral acceleration and $\kappa(s)$ as the curvature
at $s$. The speed limit of the curvature [17] is

$$\varepsilon_{\text{max}} \leq \varepsilon_{\text{max}} = \sqrt{\varepsilon_{\text{max}}^2 / \kappa(s)}.$$  

(10)

C. Forward Expansion

We present the whole process in Algorithm 1 and a demonstra-
tion in Fig. 5(a). During the expansion, the best node popped
from the open list (Algorithm 1, Line 4) is forwarded by

$$v^c = \sqrt{(v^i)^2 + 2a \cdot \Delta s}, t^c = t^i + \frac{v^c - v^i}{a},$$

(11)

where $\Delta s = s_{i+1} - s_i$, and $q^c \equiv (s_{i+1}, v, t)^c$ is the child node. $v^c \in (\varepsilon, \bar{v})(t_{i+1}, (\varepsilon, \bar{v})(t_{i+1})$ are the speed bounds at layer $l + 1$. $a$ is the acceleration from a set of discretized control inputs $\mathcal{U} = \{a_1, \ldots, a_n \in \mathbb{R}\}$. During the expansion, any child node which exceeds the layer bound $k$ or time limit $t_h$, or has a speed that is lower than $c_v$ will not be expanded (Algorithm 1, Line 6). The expanded node has cost

$$J = J_p + w_a J_a + w_v J_v,$$

(12)

where $J_p$ denotes the cost of the parent node, $J_a = u^2 \Delta t$ is the
control cost, $J_v = |v - \bar{v}|$ is the deviation from the speed limit,
and $w(\cdot)$ is the weights.

D. Local Truncation

As illustrated in Fig. 5(a), for nodes in a discretized $(v, t)$ grid
with resolutions $R_v$ and $R_t$, only the node with the lowest cost
is allowed to undergo expansion (Algorithm 1, Line 10). This
operation significantly improves the search efficiency, and it has
little impact on the search result since the abandoned nodes have
similar values to the lowest-cost one.
VI. INTERACTION PRIORITY DETERMINATION

Once the graph search is finished, trajectories to graph nodes \( V \) and the lowest-cost speed profile \( q_{1:f} = \{q^1, q^2, \ldots, q^f\} \) are obtained. For node \( q^i \) in \( q_{1:f} \), we use notation \( i \) to represent both the node and layer index for convenience. The next step is to determine which trajectory to perform, and the IPM is applied for priority checking among the sampled trajectories.

A. Speed Profile Selection

The whole process is outlined in Algorithm 2. \( V_{0:i} \) denotes the graph nodes from layer 0 to \( i \), and the IPM is used to check the priority of \( q_{1:f} \) in turn until it reaches the maximum layer \( f \) or quits when the AV’s priority is lower than that of other participants (Algorithm 2, Line 4):

\[
\exists P_i(\Delta t < 0|q^i, V_x) < C_p, \forall V_x \in A_i,
\]

where \( A_i \) is the set of agents who intend to pass the interaction point at layer \( i \). \( P_i \) denotes the probability in (9), and \( C_p \) is a constant threshold. In the implementation, protection times \( \Delta t^- \) and \( \Delta t^+ \) are calculated by setting \( \Delta t \) to zero since it is hard to estimate \( \Delta t \) accurately. Moreover, \( \min(t^+_x) \) in \( M^- \) uses the \( t \) value in \( q^i \), and \( M^+ \) is calculated via the constant acceleration model. Following the illustration in Fig. 5(b), once \( q_f \) has a lower priority, nodes at previous layers (e.g., \( s_{f-1} \)) will be checked to see if they meet the following condition

\[
(s, v, t)_l \in S(t_l), l < f,
\]

where \( S() \) is the invariable safe set [26] that at least one action exists for the AV to remain safe for an infinite time horizon. Our work involves two kinds of invariable safe sets. One is to promise the AV can stop before the intersection (Time-to-Brake [30] greater than 0), and another is to obey the responsibility safety criterion [6] when following a car. Then, a one-step forward expansion will be conducted at \( s_{f-1} \), and the generated child should obey its parent’s constraint (14). For example, as the green edges show in Fig. 5(b), child nodes at the new layer \( s_f \) are sampled if they allow the AV to stop before a safe distance to the intersection.

Finally, the best speed profile \( q_{1:f} \) is picked up, and the above operations are executed iteratively (Algorithm 2, Line 12) until all nodes of the speed profile have a higher interaction priority. Fig. 5(b) shows three speed profiles in orange, purple and grey, respectively. The orange one is the best speed profile derived from the forward simulation while having a low priority at \( p_2 \).

As a result, the speed profile in purple is selected instead. Then, the grey speed profile will be applied if the purple speed profile has a low priority at \( p_1 \).

B. Speed Profile Smoothing

The chosen speed profile \( q_{1:f} \) can be smoothed further to meet the kinodynamic feasibility using a quintic piecewise Bézier curve in the spatio-temporal semantic corridor (SSC [13]). Based on the \( \{s_1, s_2, \ldots, s_k\} \) and speed limits in the search space, our method use \( q_{1:f} \) as the initial seeds to build SSC, while dynamic obstacles are not mapped to the SSC because the initial seeds have guaranteed the AV’s safety.

VII. EXPERIMENTAL RESULTS

We train and test the IPM in the INTERACTION dataset [27] because it contains motions of various traffic participants in various driving scenarios without traffic lights. The proposed planner is validated in the CARLA [31].

A. Implementation Details

We use Pytorch [32] to train the MLP network in IPM, and the whole planning framework is implemented with C++ backend. All experiments are conducted on a computer with an AMD-5600X CPU (3.7 GHz). From the INTERACTION dataset [27], we first extract the data that have coincident trajectories between agents (as plotted in Fig. 4) and divide them into different pairs of interactions. Some examples are available on our project website, where agents are simplified as circles. At the same
interaction point, agent \( m \) overtaking \( n \) is identical to the case where agent \( n \) gives way to \( m \). As these two cases are essentially the same scenario viewed from different perspectives, we only add this type of data to our dataset once to avoid data duplication.

In the simulation, the AV’s acceleration limits are \((-3.0, 1.5)\) m/s\(^2\), and its field of view is limited, where other traffic agents will not be observed if their distance to the AV is greater than \(120 \cdot \max(1.0 - 2 \cdot |\arctan(\Delta y, \Delta x)|/\pi, 0.2)\) m. In the IPM, the MLP network and parameters \( C(\cdot) \) in (3) are available on our project website. For each intersection with a limited field of view, we apply the heuristic function \( \text{Dist}(p_o, p_j) = 0.15 \times L \) to roughly estimate the speed limit in (5), where \( L \) is the length of the lane in the intersection. During forward simulation, the maximal distance between two sampled path points is 10 m. In Algorithm 1, the expansion conditions \( t_h = 10 \) s, \( c_v = 0.1 \) m/s, grid resolutions \( R_s = 0.2 \) m/s, and \( R_t = 0.2 \) s. In the classifier (9), \( C_r = 0.5, C^2_r = 1.5, C^2 = 5.0, \) and \( C_p = 0.95 \).

### B. Interaction Protection Time

We analyze the extracted pairs of interactions (mentioned in Section VII-A) and plot \( P(\Delta t|\Delta \theta) \) and \( P(\Delta t|\Delta v) \) in Fig. 6(a). Inside a box, the green line represents the mean value, and the orange line denotes the median. The color of the box represents the sample size, where orange means the size is greater than 10000, yellow (than 1000), blue (than 100), light blue (than 10), and black is \([1, 10)\). In addition, the solid lines are fitting curves of \( \Delta t_0^-(\cdot) \) and \( \Delta t_0^+(\cdot) \) in (3), which are based on the Gaussian 3\( \sigma \) intervals.

### C. Interaction Priority

As shown in Fig. 6(b), we label samples with negative \( \Delta t \) as overtaking (red points) and positive as giving way (cyan points) in the dataset. The background color denotes MLP(\( \Delta t < 0|M^-, M^+\) in (9), and there are four areas. For B and C, the priority is clear that can be quickly classified. In A and D, \( M^+ \) plays a key role. Most agents in the dataset can give way to their competitors when the giving way ability is higher. We further classify MLP(\( \Delta t < 0|M^-, M^+ \geq 0.5 \) as overtaking to evaluate the prediction accuracy. The trained MLP network achieves 99.79% accuracy in the training set with a data size of 5283 and 99.81% in the validation set (10071 data size). In the validation experiment, we test all time slots before reaching the interaction point, which makes the validation set’s size larger than the training set’s.

### D. Advance Deceleration for Emergency

Besides the response time factor, the IPM explains why an advance deceleration is necessary for an emergency from another perspective. As shown in Fig. 6(c), the purple line represents the braking speed limit, which is the maximal speed when the braking distance is less than \( d^*_s \), and the dashed lines are the speed limits defined by (5). This figure suggests that when \( \Delta t_p < 2.0 \) s and the AV runs at 8.0 m/s, the braking is hard to prevent a collision if a vehicle suddenly appears within 16 m. That is, it reflects the speed limits of the grey area.

### E. Qualitative Results

To verify that the proposed planning framework is competent in highly interactive scenarios, we test our method in Town02 and Town04 of CARLA. In the experiments, traffic agents in the simulation are set to ignore the traffic signals and signs, and two simulation results are discussed in Fig. 7. In addition, more qualitative results are available in the supplementary video on the project website.

### F. Quantitative Results

1) **Real-time Performance:** A 30-minute navigation test with dynamical agents is conducted in CARLA to verify the

| Time (ms) | Percentage |
|----------|------------|
| <1       | 10.97%     |
| 1-10     | 42.53%     |
| 10-20    | 39.31%     |
| 20-30    | 7.16%      |
| 30-40    | <0.03%     |
computational efficiency of our proposed planning framework. The test results are shown in Table I, where the computation time includes all planning modules in Fig. 3, and the average calculation time is 10.24 ms. We observe that in 92.8% of cases, the runtime of our proposed method is less than 20 ms, which shows the real-time performance.

2) Performance Comparisons: We select Town02 as the test scenario because the map is relatively small and more suitable for comparing the interaction performance. In addition, the position of interaction points in Town02 is fixed so that we do not need to predict other agents’ paths in advance. Like previous experiments, all junctions and intersections on the map are unprotected by the traffic rules. The following algorithms are compared with the same initial location and different random seeds.

- **SSC** [13]: constant velocity (CVel) model is applied to generate the SSC.
- **MCTS** [10]: Monte Carlo tree search algorithm, where other agents are predicted to slow down to stop or speed up to a certain speed when tracking the path, and their goal probabilities are uniform.
- **MMFN-Expert** [33]: an elaborated expert agent.
- **PD-M** \[−\]: this method is based on the proposed planning framework, but the priority determination module decides to overtake when $M^−$ is less than zero.
- **PD-CVel**: like PD-M\[−\], it overtake when the $\Delta t$ estimated by CVel is less than zero.
- **PD-IPM**: the proposed planning framework using (9) to determine the priority. The speed constraint (5) is enforced in all methods since most benchmarks [10], [13] did not model the occluded factors.

Each algorithm is tested with a series of goals in five minutes. The tests are repeated five times, and the performances are recorded in Table II. In the table, the numbers are the mean values and standard deviations for three metrics: completion distance, number of collisions, and average speed. The experimental results demonstrate that our method is safe and efficient. In the seed 1 setting, PD-IPM achieves no collision situations with the longest completion length. In the seed 2 setting, although the MMFN-Expert has the longest length and highest average speed, it is unsafe and has a high collision number in red. By contrast, our PD-IPM gets a zero-collision rate and outperforms other approaches [10], [13]. Besides this, the performance of PD-M\[−\] is conservative but safe, which shows our planning framework’s reliability. The comparison between PD-CVel and PD-IPM reflects that the IPM can make the planning system more effective.

VIII. CONCLUSION

This paper presented a novel unified planning framework based on the proposed interaction point model (IPM), which enabled a uniform description of the interaction under various driving scenarios. First, the IPM was trained and tested in the...
real traffic data, showing the general interaction rule in different scenarios. Next, the planning framework was introduced, and the efficient graph search and interaction priority determination were elaborated. Experiments showed that our framework enabled the system to interact safely with other agents, and the IPM made it more efficient. Future work will include introducing more complicated scenarios into the planning framework, with interactions in open space, with pedestrians, or between other agents under more traffic regulations modeled.

REFERENCES

[1] W. Zhan, J. Chen, C.-Y. Chan, C. Liu, and M. Tomizuka, “Spatially-partitioned environmental representation and planning architecture for on-road autonomous driving,” in Proc. IEEE Intell. Veh. Symp. (IV), 2017, pp. 632–639.

[2] F. Gao, W. Wu, Y. Lin, and S. Shen, “Online Safe Trajectory Generation for Quadrrotors Using Fast Marching Method and Bernstein Basis Polynomial,” in Proc. IEEE Int. Conf. Rob. Autom., 2018, pp. 344–351.

[3] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, “Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction,” IEEE Trans. Intell. Veh., vol. 3, no. 1, pp. 5–17, Mar. 2018.

[4] L. Zhang, W. Ding, J. Chen, and S. Shen, “Efficient uncertainty-aware decision-making for automated driving using guided branching,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 3291–3297.

[5] G. Markula et al., “Defining interactions: A conceptual framework for understanding interactive behaviour in human and automated road traffic,” Theor. Issues Ergonom. Sci., vol. 21, no. 6, pp. 728–752, 2020.

[6] P. Koopman, B. Oysk, and J. Weast, “Autonomous vehicles meet the physical world: RSS, variability, uncertainty, and proving safety,” in Proc. Int. Conf. Comp. Saf., Rel., Secur, Springer, 2019, pp. 245–253.

[7] M. Makridis, K. Mattas, and B. Cuffo, “Response time and time headway of an adaptive cruise control: an empirical characterization and potential impacts on road capacity,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 4, pp. 1677–1686, Apr. 2020.

[8] Y. Jeong and K. Yi, “Target vehicle motion prediction-based motion planning framework for autonomous driving in uncontrolled intersections,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 1, pp. 168–177, Jan. 2021.

[9] L. Sun et al., “Domain knowledge driven pseudo labels for interpretable goal-conditioned interactive trajectory prediction,” 2022, arXiv:2203.15112.

[10] J. P. Hanna et al., “Interpretable goal recognition in the presence of occluded factors for autonomous vehicles,” in Proc. IEEE/RJS Int. Conf. Intell. Rob. Syst., 2021, pp. 7044–7051.

[11] W. Lim, S. Lee, M. Sunwoo, and K. Jo, “Hybrid trajectory planning for autonomous driving in on-road dynamic scenarios,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 1, pp. 341–355, Jan. 2021.

[12] T. Zhang, M. Fu, W. Song, Y. Yang, and M. Wang, “Trajectory planning based on spatio-temporal map with collision avoidance guaranteed by safety strip,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 2, pp. 1030–1043, Feb. 2022.

[13] W. Ding, L. Zhang, J. Chen, and S. Shen, “Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor,” IEEE Robot. Automat. Lett., vol. 4, no. 3, pp. 2997–3004, Jul. 2019.

[14] Y. Zhang and P. A. Ioannou, “Combined variable speed limit and lane change control for highway traffic,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 7, pp. 1812–1823, Jul. 2017.

[15] Y. Rahmati, M. K. Hosseini, and A. Talebpour, “Helping automated vehicles with left-turn maneuvers: A game theory-based decision framework for conflicting maneuvers at intersections,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 11877–11890, Aug. 2022.

[16] A. Li, L. Sun, W. Zhan, M. Tomizuka, and M. Chen, “Prediction-based reachability for collision avoidance in autonomous driving,” in Proc. IEEE Int. Conf. Rob. Autom., 2021, pp. 7908–7914.

[17] J. Cheng, Y. Chen, Q. Zhang, L. Gan, C. Liu, and M. Liu, “Real-time trajectory planning for autonomous driving with gaussian process and incremental refinement,” in Proc. IEEE Int. Conf. Rob. Autom., 2022, pp. 8999–9005.

[18] J. C. et al., “MPNP: Multi-policy neural planner for urban driving,” in Proc. IEEE/RJS Int. Conf. Rob. Autom., 2022.

[19] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, “Pip: Planning-informed trajectory prediction for autonomous driving,” in Proc. Eur. Conf. Comp. Vis. Springer, 2020, pp. 598–614.

[20] R. Xiao, X. Liu, B. Zheng, D. Liang, and Q. Zha, “Tae: A semi-supervised controllable behavior-aware trajectory generator and predictor,” 2022, arXiv:2203.01261.

[21] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, “Game-theoretic planning for self-driving cars in multivehicle competitive scenarios,” IEEE Trans. Robot., vol. 37, no. 4, pp. 1313–1325, Aug. 2021.

[22] R. Chandra and D. Manocha, “Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging,” IEEE Robot. Automat. Lett., vol. 7, no. 2, pp. 2676–2683, Apr. 2022.

[23] Y. Chen, U. Rosolia, W. Ubellacker, N. Cosmay-Shanklin, and A. D. Ames, “Interactive multi-modal motion planning with branch model predictive control,” IEEE Robot. Automat. Lett., vol. 7, no. 2, pp. 5365–5372, Apr. 2022.

[24] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, “Lookout: Diverse multi-future prediction and planning for self-driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 16107–16116.

[25] C. Pek and M. Althoff, “Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization,” in 21st Int. Conf. Intell. Transp. Syst., 2018, pp. 1447–1454.

[26] C. Pek and M. Althoff, “Fail-safe motion planning for online verification of autonomous vehicles using convex optimization,” IEEE Trans. Robot., vol. 37, no. 3, pp. 798–814, Jun. 2021.

[27] W. Zhan et al., “Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps,” 2019, arXiv:1910.03088.

[28] H. Huettenrauch, K. S. Eklundh, A. Green, and E. A. Topp, “Investigating spatial relationships in human-robot interaction,” in Proc. IEEE/RJS Int. Conf. Rob. Syst., 2006, pp. 5052–5059.

[29] M. Lee, K. Jo, and M. Sunwoo, “Collision risk assessment for possible collision vehicle in occluded area based on precise map,” in Proc. IEEE Int. Conf. Intell. Transp. Syst., 2017, pp. 1–6.

[30] J. Hildenbrand, A. M. Speier, and K. Kroschel, “A multilevel collision mitigation approach–its situation assessment, decision making, and performance tradeoffs,” IEEE Trans. Intell. Transp. Syst., vol. 7, no. 4, pp. 528–540, Dec. 2006.

[31] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in Proc. IEEE Conf. Rob. Learn., 2017, pp. 1–16.

[32] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” Proc. 33rd Int. Conf. Neural Inf. Process. Syst., vol. 7, no. 4, pp. 8026–8037, 2019.

[33] Q. Zhang, M. Tang, R. Geng, F. Chen, R. Xin, and L. Wang, “MMFN: Multi-modal fusion net for end-to-end autonomous driving,” in Proc. IEEE/RJS Int. Conf. Rob. Syst., 2022.