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

Yingbing Chen[1], Ren Xin[1], Jie Cheng[2], Qingwen Zhang[1], Xiaodong Mei[2], Ming Liu[3], and Lujia Wang[2]

***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

A. Motivation

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 scenarios, 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

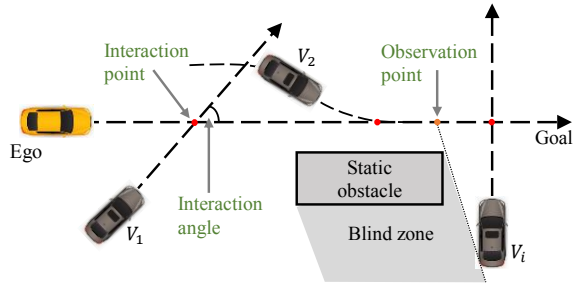


Fig. 1. The AV (ego) follows the path to the goal with considerations of interactions with other traffic agents (e.g., V_i) at different scenarios (different interaction angles and the blind zone).

"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 (Δt in Fig. 2a) 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* 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.

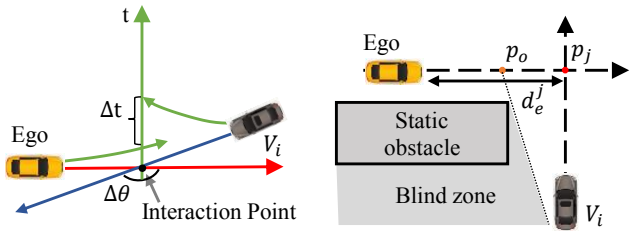
This work was supported by Guangdong Basic and Applied Basic Research Foundation, under project 2021B1515120032, Zhongshan Science and Technology Bureau Fundunder project 2020AG002, and Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone(HZQB-KCZYB-2020083), awarded to Prof. Ming Liu. (*Corresponding author: Ming Liu.*)

[1]Yingbing Chen, Ren Xin, and Qingwen Zhang are with Robotics Institute, the HKUST, Hong Kong SAR, China, the Clear Water Bay Institute of Autonomous Driving, and also with the Systems Hub, HKUST (GZ). {ychengz, rxin, qzhangcb}@connect.ust.hk.

[2]Jie Cheng, Xiaodong Mei, and Lujia Wang are with Robotics Institute, the HKUST, Hong Kong SAR, China. {jchengai, xmeiab}@connect.ust.hk, eewanglj@ust.hk.

[3]Ming Liu is with Robotics Institute, the HKUST, Hong Kong SAR, China, the HKUST (Guangzhou), Nansha, Guangzhou, 511400, Guangdong, China, and also with the HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen. eelium@ust.hk.

*IPM model files and demonstration videos are available on the project website: https://github.com/ChenYingbing/IPM-Planner.



(a) Demonstration of $\Delta\theta$ and Δt (b) Area with limited FoV

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

B. 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 searching to calculate the trajectory distribution under traffic regulations, and a priority determination module based on the IPM to select a safe trajectory.
- 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]–[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 plannings [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]–[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 plannings [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].

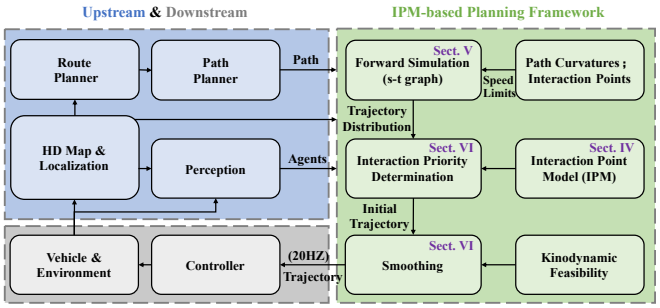


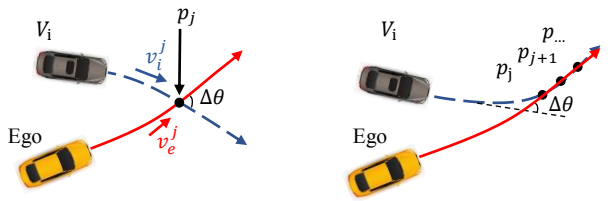
Fig. 3. Overview of the proposed planning framework with IPM and its relationship with other AV system components.

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].



(a) Point-Overlap (b) Line-Overlap

Fig. 4. Illustrations of the interaction point p, angle $\Delta\theta$, and speed v. (b) shows that the interaction angle should consider the shapes of the agents.

It should be noted that the proposed IPM is also applicable in one-to-many interactions.

A. Assumptions

The human-robot interaction model [28] indicates that participants will exchange their intentions beyond a certain distance called the social distance. Within the social distance, the determined form of interaction is performed. Similarly, [5] points out that, before an interaction happens, traffic participants show their intentions via implicit (interactive behavior) or explicit (e.g., turn signal) communications. Inspired by these works, we make the following assumptions:

Assumption 1. The participants will be conservative when facing potential interactions until the maneuver (e.g., who goes first) is determined via communication [5].

Assumption 2. The maneuver for an interaction point can be determined only after all participants are observed. There is a location called an observation point at which there is no blind zone in FoV.

For *Assumption 1*, we believe it is rational because even in a car-following scenario [6], traffic agents act conservatively to keep a safe distance from the front vehicle. That is, the planned trajectory should guarantee the safety of the AV for the worst-case situations in the future [26]. As for *Assumption 2*, an example of the observation point is represented in Fig. 2b, where the AV cannot determine its maneuver to interact with the V_i until it reaches the position of the observation point p_o. How to obtain an observation point is not discussed in this work, it requires an analysis of the FoV and high-definition map [29].

B. Interaction Point

The properties of an interaction point are illustrated in Fig. 4a, where p_j denotes the j-th interaction point, and v_e^j and v_i^j are the speeds of the moment that agents pass the p_j. We further define

$$\Delta v = v_e^j - v_i^j, \Delta\theta = |\theta_e^j - \theta_i^j|, \Delta t = t_e^j - t_i^j, \tag{1}$$

where Δv and $\Delta\theta$ are the relative interaction speed and angle, and Δt is the time gap in Fig. 2a. In the equation, similar to the definition of v_e^j, θ_e^j is the yaw state of the AV at the location of p_j, and t_e^j is the arrival time. The subscript letter indicates the agent, and the superscript corresponds to the index of the interaction point. Moreover, Δt is nonzero because agents cannot pass the same position simultaneously.

C. Interaction Protection Time

Following *Question 1*, the interaction protection time is the minimum value of the time gap Δt. It has two cases: A negative Δt corresponds to overtaking, and a positive one corresponds to yielding for others. As a result, we define

$$\begin{cases} \Delta t_p^- = \max \Delta t, \forall \Delta t \in \mathcal{T} \cap \mathbb{R}^-, \\ \Delta t_p^+ = \min \Delta t, \forall \Delta t \in \mathcal{T} \cap \mathbb{R}^+, \end{cases} \tag{2}$$

where Δt_p^- denotes the maximum time gap when overtaking, Δt_p^+ is the minimum time gap when giving way, and $\mathcal{T}$ is the set of possible values of the time gap in the data set. Then, $\Delta t_p^{(\cdot)}$ is modeled as functions of $\Delta\theta$ and Δv respectively.

$$\begin{aligned} \Delta t_p^-(\Delta v) &= \mathbf{C_1} \begin{bmatrix} \Delta v^4 & \Delta v^3 & \Delta v^2 & \Delta v & 1 \end{bmatrix}^T, \\ \Delta t_p^-(\Delta \theta) &= \mathbf{C_2} \begin{bmatrix} \Delta \theta^2 & \Delta \theta & 1 \end{bmatrix}^T, \\ \Delta t_p^+(\Delta v) &= \mathbf{C_3} \begin{bmatrix} \Delta v^4 & \Delta v^3 & \Delta v^2 & \Delta v & 1 \end{bmatrix}^T, \\ \Delta t_p^+(\Delta \theta) &= \mathbf{C_4} \begin{bmatrix} \Delta \theta^2 & \Delta \theta & 1 \end{bmatrix}^T, \end{aligned} \tag{3}$$

where $\Delta t_p^{(\cdot)}(\Delta v)$ and $\Delta t_p^{(\cdot)}(\Delta\theta)$ are represented as fitting curves, and $\mathbf{C}_{(\cdot)}$ is the fitting coefficient. Given the values of Δv and $\Delta\theta$ together, the boundaries become

$$\begin{cases} \Delta t_p^- = \min\left(\Delta t_p^-(\Delta v), \Delta t_p^-(\Delta\theta)\right), \\ \Delta t_p^+ = \max\left(\Delta t_p^+(\Delta v), \Delta t_p^+(\Delta\theta)\right). \end{cases} \tag{4}$$

More illustrations will be given in Sect. 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. Fig. 2b shows a demonstration, where before the AV reaches the observation point p_o, the remaining time T to reach p_j should be greater than Δt_p^+. This corresponds to the speed upper bound

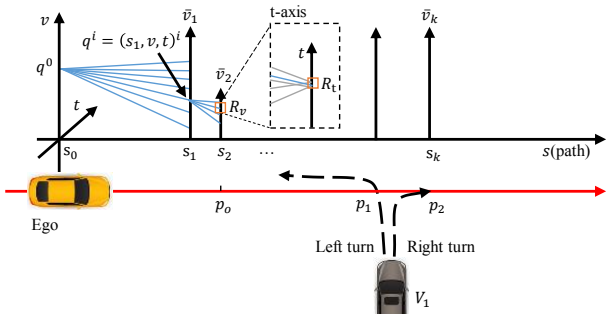
$$\overline{v} = \frac{d_e^j}{T} \leq \frac{d_e^j}{\Delta t_p^+}, d_e^j \geq \text{Dist}(p_o, p_j), \tag{5}$$

where d_e^j is the distance between the AV and the p_j, $\text{Dist}(\cdot)$ returns the Euler distance, and the upper bound meets its minimum $\overline{v} = \text{Dist}(p_o, p_j)/\Delta t_p^+$. In this equation, an estimation of Δt_p^+ can be quickly obtained by setting Δv to zero, which means there is no need to predict agents' speed in the occluded area.

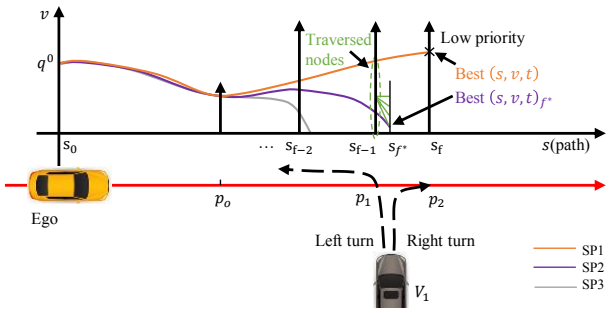
E. Interaction Priority

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

$$\min(t_i^j) = \frac{\sqrt{v_{0i}^2 + 2\max(a_i) d_i^j} - v_{0i}}{\max(a_i)}, \tag{6}$$



(a) Expansion and local truncation



(b) Speed profile generation

Fig. 5. Illustrations of the efficient s-t graph search and speed profile generation. (a) Results of the forward expansion for two rounds, where the grey lines in the t-axis are abandoned edges during local truncation. (b) Speed profiles (as colored lines) are generated in the interaction priority determination.

$$\max\left(t_i^j\right) = \frac{v_{0i} - \sqrt{v_{0i}^2 + 2\min(a_i)d_i^j}}{|\min(a_i)|}, \quad \text{s.t. } a_i \geq -v_{0i}^2/(2 \cdot d_i^j), \tag{7}$$

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 acceleration and deceleration. 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

$$\begin{aligned} M_j^- &= \min(t_e^j) - \min(t_i^j) + |\Delta t_p^-|, \\ M_j^+ &= \max(t_e^j) - \max(t_i^j) - |\Delta t_p^+|, \end{aligned} \tag{8}$$

where the expression of M^- indicates that the overtaking needs to suffer the protection time Δt_p^-, 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 overtake, the communications mentioned in *Assumption 1* and the traffic regulations 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

$$\begin{cases} P_j(\Delta t < 0) = 0.0, M^- \geq t_m, \\ P_j(\Delta t < 0) = \text{MLP}(M^-, M^+), M^- < t_m, \end{cases} \tag{9}$$

where MLP is the trained 3-layer MLP model, $t_m = \min(C_r^1 \cdot t_e^j + C_r^2, C_r^3)$ is the time threshold, and $C_r^{(\cdot)}$ are constants (C_r^1 is positive). When the AV is far from the interaction point (t_e^j is large), it is assumed that the interaction follows patterns learned from the dataset (M^- is less than t_m). In contrast, when t_e^j 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 acceleration 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 $q^i = (s_l, v, t)^i$, where i is the node index, $s_{(\cdot)}$ is the accumulated distance, the subscript l is called the layer index, and v and t are the speed and timestamp values.

Algorithm 1 S-t graph search

Notation: open list $\mathcal{O}$, layer indexs l and f, path points $\{s_1, s_2, ..., s_k\}$, node list $\mathcal{V}$, the maximum layer k, constants t_h and c_v
Initialize: $\mathcal{O} \leftarrow \{q^0\}$, $\mathcal{V} \leftarrow \emptyset$, $f \leftarrow 0$
for $n_{iter} \in \{0, \ldots, \text{max\_iter}\}$ **do**
  $q^i \equiv (s_l, v, t)^i \leftarrow$ PopBestNode($\mathcal{O}$)
  $f \leftarrow \max(f, l)$, $\mathcal{V}_l \leftarrow \mathcal{V}_l \cup q^i$
  if OutOfBounds(q^i, l, k, t_h, c_v) **then**
    Continue
  end if
  for $a \in \{a_1, \ldots, a_n\}$ **do**
    $q^c \leftarrow$ ForwardExpansion(q^i, a, s_l, s_{l+1})
    $J =$CalculateCost(q^i, q^c)
    if LocalTruncation(q^c, J, R_v, R_t) **then**
      $\mathcal{O} \leftarrow \mathcal{O} \cup q^c$
    end if
  end for
end for
$q^* \leftarrow$ BestNode($\mathcal{V}_f$)
return $\mathcal{V}$, PathTo(q^*)

A. Path Point Selection

The first step is to select a sequence of path points $\{s_1, s_2, ..., s_k\}$ to build the search space, where k is the maximum 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 straights. (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 computation 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). The speed constraint of the curvature follows [17]

$$\overline{v}_\kappa \leq v_\kappa^{\max} = \sqrt{a_{\text{lat}}^{\max}/|\kappa(s)|}, \tag{10}$$

where $a_{\text{lat}}^{\max}$ is the maximum lateral acceleration, and $\kappa(s)$ is the curvature.

C. Forward Expansion

We present the whole process in Algorithm 1 and a demonstration in Fig. 5a. During the expansion, the best node $(s_l, v, t)^i$ from the open list (Algo. 1, Line 4) is forwarded by the following equation:

$$v^c = \sqrt{(v^i)^2 + 2a \cdot \Delta s}, t^c = t^i + \frac{v^c - v^i}{a}, \tag{11}$$

where $\Delta s = s_{l+1} - s_l$, and $q^c \equiv (s_{l+1}, v, t)^c$ is the child node. $v^c \in (\underline{v}, \overline{v})_{l+1}$, $(\underline{v}, \overline{v})_{l+1}$ are the speed bounds at layer $l+1$. a is the acceleration from a set of discretized control inputs $\mathcal{U} = \{\underline{a} \leq a_1, \ldots, a_n \leq \overline{a}\}$. 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 (Algo. 1, Line 6). The expanded node has cost

$$J = J_p + w_a J_a + w_v J_v, \tag{12}$$

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

D. Local Truncation

As illustrated in Fig. 5a, 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 (Algo. 1, Line 12). 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 best-cost one.

VI. Interaction Priority Determination

Once the graph search is finished, trajectories to graph nodes $\mathcal{V}$ and the best-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. $\mathcal{V}_{0:i}$ 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 (Algo. 2, Line 4):

$$\exists P_i(\Delta t < 0 | q^i, V_x) < C_p, \forall V_x \in \mathcal{A}_i, \tag{13}$$

Algorithm 2 Priority Determination

1: **Notation**: agent list $\mathcal{A}$
2: **Initialize**: $\mathcal{V}, q_{1:f} \leftarrow$ StGraphSearch()
3: **for** i in $\{1, \ldots, f\}$ **do**
4: **if** PriorityIsLow$(q^i, \mathcal{A}_i)$ **then**
5: $q_{1:f^*} \leftarrow$ CheckAndExpand$(\mathcal{V}_{0:i-1})$
6: **break**
7: **end if**
8: **end for**
9: **if** $q_{1:f^*} ==$ None **then**
10: **return** Smooth$(q_{1:f})$
11: **end if**
12: **return** PriorityDetermination$(\mathcal{V}, q_{1:f^*})$

where $\mathcal{A}_i$ 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 Δt_p^- and Δt_p^+ are calculated by setting Δv to zero since it is hard to estimate Δv accurately. Moreover, $\min(t_e^i)$ in M^- uses the t value in q^i, and M^+ is calculated via the constant acceleration model. Following the illustration in Fig. 5b, 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

$$(s, v, t)_l \in S(t_l), l < f, \tag{14}$$

where $S(\cdot)$ 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. 5b, 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 (Algo. 2, Line 12) until all nodes of the speed profile have a higher interaction priority. Fig. 5b 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 [13]. Based on the $\{s_1, s_2, ..., s_k\}$ and speed limits in the search space, we use $q_{1:f^*}$ as the initial seeds to build SSC. Dynamic obstacles are not mapped to the SSC because the initial seeds have guaranteed the AV's safety. For more details, please refer to [13].

TABLE I

TIME CONSUMPTION OF PLANNING FRAMEWORK

Time	<1ms	1-10ms	10-20ms	20-30ms	30-40ms
Percentage	10.97%	42.53%	39.31%	7.16%	<0.03%

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.

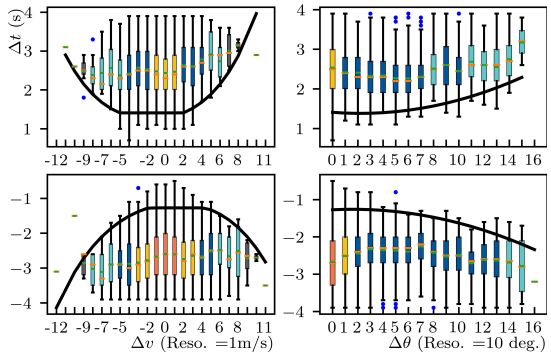
In the simulation, the AV's acceleration limits are $(-3.0, 1.5)$ m/s^2, and its field of view is limited, where other traffic agents will not be observed if their distance to the AV is greater than $120 \cdot \max(1.0 - 2 \cdot |\arctan(\Delta y, \Delta x)|/\pi, 0.2)$ m. In the IPM, the MLP network and parameters $\mathbf{C}_{(\cdot)}$ 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 $10m$. In Algorithm 1, the expansion conditions $t_h = 10$ s, $c_v = 0.1$ m/s, grid resolutions $R_v = 0.2$ m/s, and $R_t = 0.2$ s. In the classifier (9), $C_r^1 = 0.5$, $C_r^2 = 1.5$, $C_r^3 = 5.0$, and $C_p = 0.95$.

B. Interaction Protection Time

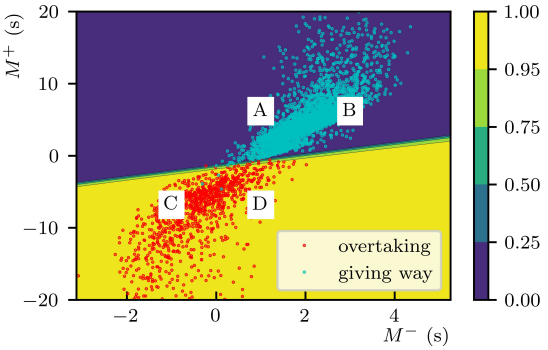
We analyze the extracted pairs of interactions (mentioned in Sect. VII-A) and plot $P(\Delta t|\Delta\theta)$ and $P(\Delta t|\Delta v)$ in Fig. 6a. Inside a box, the green line represents the mean value, and the orange line denotes the median. The color of the box represents the sample size, where orange 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 $\Delta t_p^-(\cdot)$ and $\Delta t_p^+(\cdot)$ in (3), which are based on the Gaussian 3σ intervals.

C. Interaction Priority

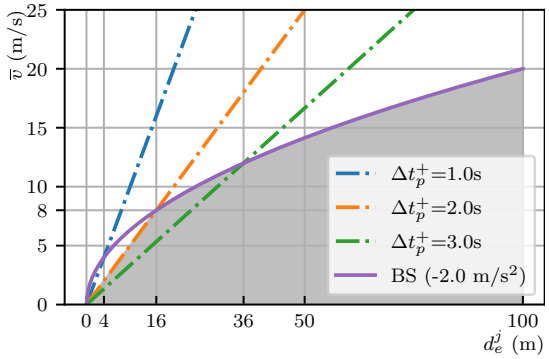
As shown in Fig. 6b, we label samples with negative Δt as overtaking (red points) and positive as giving way (cyan points) in the dataset. The background color denotes $\text{MLP}(\Delta t < 0|M^-, M^+)$ in (9), and there are four areas.



(a) Illustrations of $P(\Delta t|\Delta v)$, $P(\Delta t|\Delta\theta)$ and fitting curves.



(b) Training result of the interaction priority network, where points are labelled samples in the training set, and the background color denotes the probability calculated by MLP.



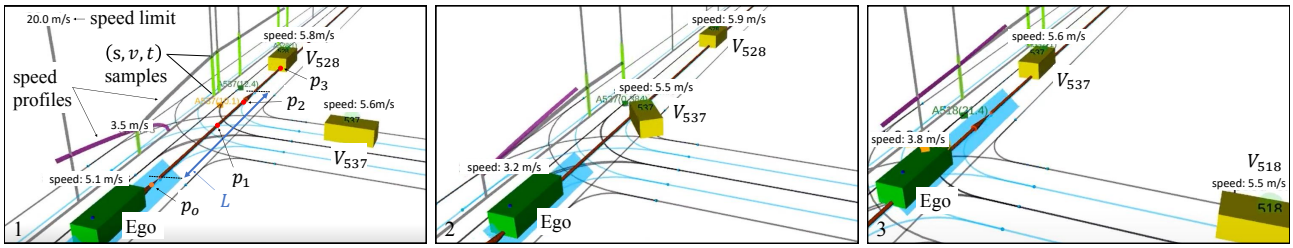
(c) Comparisons of speed limits.

Fig. 6. Experiments of the interaction point model.

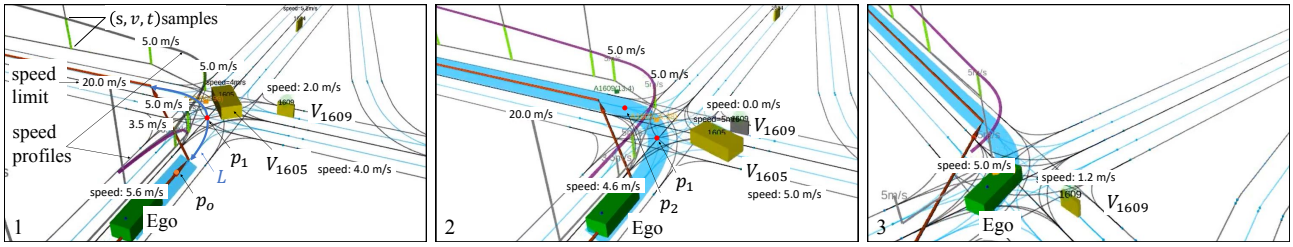
For B and C, the priority is clear that can be quickly classified. In A and D, M^+ plays a key role. Most agents in the dataset can give way to their competitors when they have a higher giving way ability. We further classify $\text{MLP}(\Delta t < 0|M^-, M^+) \geq 0.5$ as overtaking to evaluate the prediction accuracy. In the experiment, 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 causes the validation set's size to be larger than the training set's.

D. Advance Deceleration for Emergency

In addition to the response time factor, the IPM explains why an advance deceleration is necessary for an emergency



(a) Giving way to other vehicles. (1) Due to the speed limit at p_o, the AV slows down from 5.1m/s, and it has a lower priority than V_{537} at both p_1 and p_2. (2) When V_{537} turns right, the AV decides to occupy p_1. (3) Finally, the AV follows V_{537} while maintaining a certain speed.



(b) Overtaking other vehicles. (1) Firstly, the AV decides to stop before p_1. (2) Once V_{1605} turns left, the AV starts to compete with V_{1069}. Meanwhile, the AV believes it has a higher priority than V_{1609} at p_1 and p_2. (3) The AV successfully overtakes V_{1069}.

Fig. 7. Illustration of different interaction results in a conflict zone.

TABLE II

QUANTITATIVE EXPERIMENTAL RESULTS

	Town02 (Seed 1)			Town02 (Seed 2)		
Methods	Completion (m) ↑	Collision Num.↓	Avg. Speed[a](m/s)↑	Completion (m) ↑	Collision Num.↓	Avg. Speed (m/s)↑
SSC [13]	1232.00 ± 187.71	**0.0** ± 0.00	4.10 ± 0.61	1444.94 ± 081.07	0.2 ± 0.45	4.8 ± 0.25
MCTS [10]	1295.73 ± 054.82	**0.0** ± 0.00	4.43 ± 0.31	1401.86 ± 104.27	**0.0** ± 0.00	4.7 ± 0.36
MMFN-Expert [33]	1340.84 ± 412.39	1.0 ± 0.71	**4.86** ± 0.53	**1620.00** ± 059.71	0.8 ± 0.84	**5.4** ± 0.18
PD-M	1193.86 ± 107.04	**0.0** ± 0.00	3.98 ± 0.38	1303.22 ± 035.57	**0.0** ± 0.00	4.3 ± 0.11
PD-CVel	1303.08 ± 161.48	0.3 ± 0.48	4.41 ± 0.52	1398.94 ± 081.53	0.2 ± 0.45	4.7 ± 0.30
PD-IPM	**1358.11** ± 060.73	**0.0** ± 0.00	4.71 ± 0.48	1457.96 ± 042.54	**0.0** ± 0.00	4.9 ± 0.13

[a] is the average velocity before the AV collides with other vehicles and can not continue with the navigation task.

from another perspective. As shown in Fig. 6c, the purple line represents the braking speed limit, which is the maximal speed at which the braking distance is less than d_e^j, and the dashed lines are the speed limits defined by (5). This figure indicates that when Δt_p^+ is 2.0 s, and the AV runs at around 8.0 m/s, it is hard for the braking 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 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 overtakes when the Δt 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. CONCLUSIONS

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. Z. et al., "Spatially-partitioned environmental representation and planning architecture for on-road autonomous driving," in *IEEE Intell. Veh. Symp. (IV)*. IEEE, 2017, pp. 632–639.

[2] F. G. et al., "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *IEEE Int. Conf. Rob. Autom. (ICRA)*. IEEE, 2018, pp. 344–351.

[3] H. et al., "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 1, pp. 5–17, 2018.

[4] L. Z. et al., "Efficient uncertainty-aware decision-making for automated driving using guided branching," in *2020 IEEE Int. Conf. Robot. Automat.* IEEE, 2020, pp. 3291–3297.

[5] G. M. et al., "Defining interactions: A conceptual framework for understanding interactive behaviour in human and automated road traffic," *Theoretical Issues in Ergonomics Science*, vol. 21, no. 6, pp. 728–752, 2020.

[6] P. K. et al., "Autonomous vehicles meet the physical world: Rss, variability, uncertainty, and proving safety," in *Int. Conf. Comp. Safety, Reliability, Security*. Springer, 2019, pp. 245–253.

[7] M. et al., "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, 2019.

[8] J. Y. et al., "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, 2019.

[9] L. S. et al., "Domain knowledge driven pseudo labels for interpretable goal-conditioned interactive trajectory prediction," *arXiv preprint arXiv:2203.15112*, 2022.

[10] J. P. H. et al., "Interpretable goal recognition in the presence of occluded factors for autonomous vehicles," in *IEEE/RJS Int. Conf. Intell. Rob. Syst. (IROS)*. IEEE, 2021, pp. 7044–7051.

[11] W. L. et al., "Hybrid trajectory planning for autonomous driving in on-road dynamic scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 341–355, 2019.

[12] T. Z. et al., "Trajectory planning based on spatio-temporal map with collision avoidance guaranteed by safety strip," *IEEE Trans. Intell. Transp. Syst.*, 2020.

[13] W. D. et al., "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2997–3004, 2019.

[14] Y. Z. et al., "Combined variable speed limit and lane change control for highway traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1812–1823, 2016.

[15] Y. R. et al., "Helping automated vehicles with left-turn maneuvers: a game theory-based decision framework for conflicting maneuvers at intersections," *IEEE Trans. Intell. Transp. Syst.*, 2021.

[16] A. L. et al., "Prediction-based reachability for collision avoidance in autonomous driving," in *IEEE Int. Conf. Rob. Autom. (ICRA)*. IEEE, 2021, pp. 7908–7914.

[17] J. C. et al., "Real-time trajectory planning for autonomous driving with Gaussian process and incremental refinement," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2022, pp. 8999–9005.

[18] J. C. et al., "Mpnp: Multi-policy neural planner for urban driving," in *IEEE/RJS Int. Conf. Intell. Rob. Syst. (IROS)*. IEEE, 2022.

[19] H. S. et al., "Pip: Planning-informed trajectory prediction for autonomous driving," in *Eur. Conf. Comp. Vis. (ECCV)*. Springer, 2020, pp. 598–614.

[20] J. R. et al., "Tae: A semi-supervised controllable behavior-aware trajectory generator and predictor," *arXiv preprint arXiv:2203.01261*, 2022.

[21] M. W. et al., "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1313–1325, 2021.

[22] R. C. et al., "Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2676–2683, 2022.

[23] Y. C. et al., "Interactive multi-modal motion planning with branch model predictive control," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5365–5372, 2022.

[24] A. C. et al., "Lookout: Diverse multi-future prediction and planning for self-driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 16 107–16 116.

[25] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *ITSC*. IEEE, 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, 2020.

[27] Z. W. et al., "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[28] H. H. et al., "Investigating spatial relationships in human-robot interaction," in *IEEE/RJS Int. Conf. Intell. Rob. Syst. (IROS)*. IEEE, 2006, pp. 5052–5059.

[29] M. L. et al., "Collision risk assessment for possible collision vehicle in occluded area based on precise map," in *IEEE Int. conf. Intell. Transp. Sys. (ITSC)*. IEEE, 2017, pp. 1–6.

[30] J. H. et al., "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, 2006.

[31] A. D. et al., "Carla: An open urban driving simulator," in *Proc. IEEE Conf. Rob. Learn. (CoRL)*. PMLR, 2017, pp. 1–16.

[32] A. P. et al., "Pytorch: An imperative style, high-performance deep learning library," *NeurIPS*, vol. 32, 2019.

[33] Q. Z. et al., "MMFN: Multi-modal fusion net for end-to-end autonomous driving," in *IEEE/RJS Int. Conf. Intell. Rob. Syst. (IROS)*. IEEE, 2022.