2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

Decision-making and Planning Framework with Prediction-Guided Strategy Tree Search Algorithm for Uncontrolled Intersections

Ting Zhang[1], Mengyin Fu[1,2], Wenjie Song[1,*], Yi Yang[1], Meiling Wang[1]

Abstract— **Uncontrolled intersections are important and challenging traffic scenarios for autonomous vehicles. Vehicles not only need to avoid collisions with dynamic vehicles instantaneously but also predict their behavior then make long-term decisions in reaction. To solve this problem, we propose a cooperative framework composed of a Primary Driver (PD) for motion planning and a Subordinate Driver (SD) for decision-making. SD is essentially the combination of a prediction module and a high-level behavior planner, which develops a prediction-guided strategy tree to determine the optimal action sequence. Especially, under the guidance of the prediction results, the tree branches are evaluated in security metrics, then get trimmed in action and observation space to reduce the dimensional complexity. With the assistance of SD, PD works as a collision checker and a low-level motion planner to generate a safe and smooth trajectory. We use the INTERACTION dataset to validate our method and achieve more than 90% success rate with efficiency improvement in various situations.**

I. INTRODUCTION

Intersections without traffic signals have drawn much attention in autonomous driving recently, which requires more intelligent decision-making. According to [1], more than one-fourth of fatal crashes in the US are related to intersections, and about 50% of these occur in uncontrolled intersections. The interactive feature of intersections makes it necessary to estimate the future situation then make decisions accordingly, and the motion planning should be conducted iteratively to handle the uncertainties. To this end, we propose a framework that contains the 'Primary Driver (PD)' and the 'Subordinate Driver (SD)'. PD stands for the role who detects collision risk and conducts real-time motion planning by manipulating the pedal and steer wheel. SD makes decisions to assist PD, which serves as a behavior planner coupled with a prediction module. Based on the predicted multi-modal trajectories of the target vehicles, SD explores the optimal action by designing a strategy tree, named prediction-guided strategy tree (PGST). After branch trimming by the predicted states, the tree expands efficiently and brings out the action pairs with risk assessment. Our framework is hierarchically developed based on this concept and elaborates on the design of PGST.

During the process of decision-making, two issues become increasingly prominent. One is caused by the misunderstanding of other agents. For example, without accurate prediction, the ego vehicle may waste a long time waiting outside the intersection (conservative) or risk crossing (aggressive). In consequence, long-term prediction is critically important. Recently, graph-based prediction networks [2]–[4] become popular for their ability to model the interaction between agents and the environment. On top of that, 'anchor trajectories' have been proposed to enumerate the possible future trajectories, which reformulates the prediction problem into anchor selection and offset regression [5], [6]. Compared with them, target-driven model TNT [7] stands out for these advantages: 1) generate reliable trajectories by using domain knowledge like road geometry, 2) generate multi-modal trajectories with scored probability. Hence, we adopt it as the prediction model of SD.

The second problem is the unawareness of uncertainty. The uncertainties are inevitable during the perception and prediction process. Partially Observable Markov Decision Process (POMDP) provides an explicit way to model the belief of the world [8]–[10]. As the situation (search depth) evolves, the dimension of the model exponentially expands concerning the observation space and action space, making it intractable to work online. Besides, the discrete action transition leads to an unsmooth trajectory. And the limited prediction horizon (1-3 steps, less than 2s) paralyzes it from making a long-horizon plan. To this end, [11] improves POMDP by considering the constraints from the behavior and the environment and substituting the basic action with the semantic policy. Furthermore, to improve the operational efficiency and flexibility, [12], [13] propose a guided branch to reduce the complexity.

To conquer these challenges, rule-based models are combined with learning-based models to get better performance. One type of model uses the learning-based decision-maker accompanied with the rule-based controller. [14] has developed a model composed of a reinforcement learning model as the high-level decision-maker, and also an MPC controller to optimize a safe trajectory. [15] proposes a hierarchical behavior planning framework, in which a high-level reinforcement learning algorithm works as a coordinator to guide the low-level quadratic regulator-based controllers. Another type belongs to the rule-based decision-maker coupled with an end-to-end learner to generate admissible actions. Macro-Action Generator-Critic [16] propagates the actions of an online

This work was partly supported by Program for National Natural Science Foundation of China (Grant No. U1913203, 61903034, 61973034 and 91120003), Changjiang Scholars and Innovative Research Team in University (IRT-16R06, T2014224), Youth Talent Promotion project of China Association for science and technology, Beijing Institute of Technology Research Fund Program for Young Scholars.

[1]The authors are with State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, P.R.China.
[2]The author is also with Nanjing University of Science and Technology, Nanjing 210014, P.R.China.
*Corresponding author: Wenjie Song (email: wenjiebit@gmail.com).

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922066

planner (POMDP) as feedback to learn an end-to-end macro-action generator. Another closed-loop work, called LeTS-Drive [17], learns driving policies from a sparsely sampled tree search-based planner. It then uses this learned policy to guide planning for real-time vehicle control. Learning-based prediction can also contribute to the good performance of planners. In [12], the authors design a lightweight neural network for estimating the semantic action of other traffic participants, which adapts to the various environment easily. Imitation learning is also utilized with the game-theoretic model for predicting the trajectories in response to the merging ego vehicle's action [18].

Inspired by these models, we integrate the learning-based prediction with the tree search algorithm to build a prediction-guided strategy tree for decision-making, having the prediction distribution included in the security metrics to evaluate the action candidates of the tree. To alleviate the computational burden, we try to trim the branch in the following aspects:

- Target Vehicles: Narrow down the target vehicles by their predicted intentions (turning left/right, going straight) and the traveling phase (approach, inside, and departure).
- Observation Space: Predict the future trajectory distribution through the neural network, and all the possible prediction results are considered concurrently with different attention, reducing the observation branches.
- Action Pruning: the action branches composed of adaptive acceleration and semantic path candidates are pruned by the security metrics. The completely safe actions are picked out directly, while others will be kept in an action pool, waiting to be selected by the safety scores.

The main contributions could be concluded as:

1) We propose a cooperative framework for vehicles in uncontrolled intersections, which is composed of a primary driver for planning and a subordinate driver for decision-making concerning interaction and uncertainty.
2) We develop a prediction-guided strategy tree algorithm for optimal action search. Given the possible trajectory distribution of the target vehicles, the observation and action branch of the tree will get trimmed with probability, which reduces the computational complexity.
3) We validate the framework on the real-world dataset, and the results demonstrate the remarkable efficiency of our method in comparison to the human drivers.

A. Formulations

We define the static environment of the uncontrolled intersection as m, the center lines as L_m, the index of ego vehicle as 0, the index of other target vehicles as $i = 1, 2, \cdots, N$ (superscript). At the beginning time stamp t_0 of the decision horizon $T_F = 3$s, it is assumed that the current state of the ego vehicle $\mathcal{X}_{t_0}^0$ and the other vehicles $\mathcal{X}_{t_0}^{i=1,2,\cdots,N_{t_0}}$ can be obtained, where $\mathcal{X} = (x, y, v, a, s, d, \phi, \mathrm{id})$ ((x, y), (s, d) are coordinates in global frame and Frenet frame, and

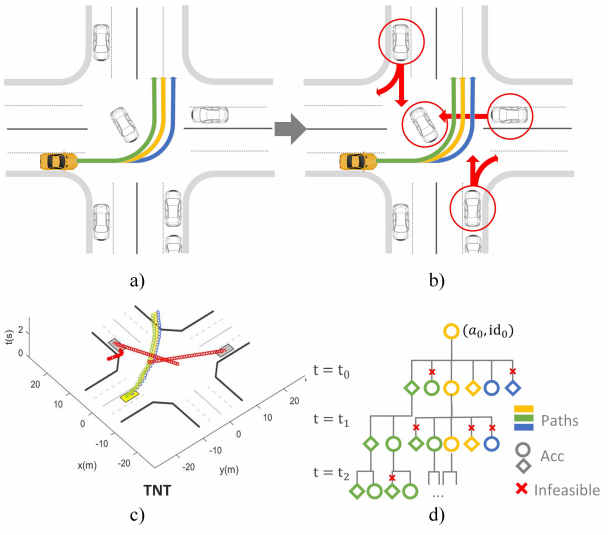


Fig. 1. Schematic diagram of the decision process in the proposed framework. a) Path candidates generation; b) Predicting the intentions of the target vehicles in red circles; c) Predicting the multi-modal trajectories of the target vehicles in spatio-temporal domain; d) Conducting the prediction-guided strategy tree search. Each node represents the action candidate composed of the acceleration (different shape) and path candidate (different color). High-risk actions with red marks are pruned.

v and a are velocity and acceleration. ϕ is the heading angle, and id refers to the index of the path candidate.). The past states in historical period T_H can be abbreviated to $\mathcal{X}_{T_H}$ (no superscript means all agents).

B. Reference Paths Generation

For the ego vehicle, the path candidates are represented by l_m^0, among which the reference path $l_{t_0}^0$ is the path it follows. The path candidates l_m^0 guide the ego vehicle through the intersection, which can be generated from the upstream module, like the route planner. Conditioned on the map or the lane boundaries, we first find out the available lanes for an exit, then samples in Frenet frame for different bias d_i in d direction. Given destinations, we deploy the spline to represent the reference paths (green, yellow, blue curves in Fig. 1), realizing the transformation from global coordinate to the Frenet coordinate.

C. Target Vehicle Selection

For vehicles in intersections, with the estimated intention of turning or going straight, it is easy to clarify the target vehicles that should be paid more attention to. On account of the various shape and sizes of the intersection, we do not pick out vehicles based on fixed distance threshold but try to figure out the target vehicles that share potential interaction with the ego vehicle. Specifically, for all vehicles within the perception range, we first classify them according to travel phases, including approaching, inside, and departure phases. For vehicles that have left the intersection, we do not consider them anymore. Vehicles inside the intersection are all regarded as target vehicles because they may hold the right-of-way or have potential conflicts. According to priorities, vehicles approaching the intersection and closest

to the stop line (or entrance) in each lane are concerned. Consequently, N_{t_0} vehicles are selected at timestamp t_0 in total, which are marked with red circles in Fig. 1.

D. Prediction of Other Vehicles

Prediction in intersections differs from that in highway environments, which should reflect on the uncertainty and maneuverability caused by interaction, including the mutual interaction among vehicles and the interaction with the static environment. Therefore, we decide to use the TNT model, which consists of three interpretable stages: target prediction, target-conditioned motion estimation, and trajectory scoring. It first uniformly samples potential targets along the center lines, then by encoding the interactions with the environment and other agents, TNT produces trajectories conditioned on targets. The scoring stage finally estimates the likelihood of the trajectories and selects the trajectories ranking top $k_{\text{TNT}} = 6$ as results.

To find out the candidate goals of the target vehicle, we trained a Random Forest (RF) model to estimate their intentions of the behaviors like going straight, turning left and right $\zeta_{t_0}^{N_{t_0}} = \{\zeta_S^n, \zeta_L^n, \zeta_R^n | n \in N_{t_0}\}$. With intentions, reference paths will be selected from L_m. Then, according to the current states $\mathcal{X}_{t_0}^{N_{t_0}}$, we infer their range of destination $(s_{\min}^{N_{t_0}}, s_{\max}^{N_{t_0}})$ with reference to the vehicle model. Consequently, goal candidates $\mathcal{G}_{tar}^{N_{t_0}}$ will be sampled with resolution r_0 in the range of each path.

II. PREDICTION-GUIDED STRATEGY TREE SEARCH

Our work is aimed at interactive and uncertainty-aware motion planning problems in the uncontrolled intersection. When the ego vehicle is approaching the entrance and following a preceding vehicle, the simple Intelligent Driver Model (IDM) [19] is used as (1), where $v_{\text{des}}, s_{\text{des}}$ are the desired velocity and distance, and v_0, v_l are the velocity of the ego and preceding vehicle. s_0 is the distance between the two vehicles and T_0 is the reaction time, and the exponent δ is usually set to 4. As shown in Fig. 2, PD is running at 10Hz to avoid obstacles in time. SD works every 1s (1Hz) with the planning horizon $T_F = 3$s aligned with the prediction horizon, which is the trade-off of the time consumption and the decision-making demand. Without any potential conflict, the quick planning will generate a smooth trajectory from the current state $\mathcal{X}_{t_0}^0$ to reach the desired velocity. Otherwise, with the k_{TNT} possible trajectories and scores, the prediction-guided tree will initiate to determine the optimal action sequence $\mathbf{a}_{T_F}^*$. Next, we will get into the details of the prediction-guided strategy tree search algorithm of SD.

$$\dot{v} = a_{\max}(1 - (\frac{v_0}{v_{\text{des}}})^{\delta} - (\frac{s(v_0, \Delta v)}{s_0})^2),$$
$$\Delta v = v_0 - v_l, s(v_0, \Delta v) = s_{\text{des}} + v_0 T_0 + \frac{v_0 \Delta v}{2\sqrt{a_{\max} a_{\text{brake}}}}. \quad (1)$$

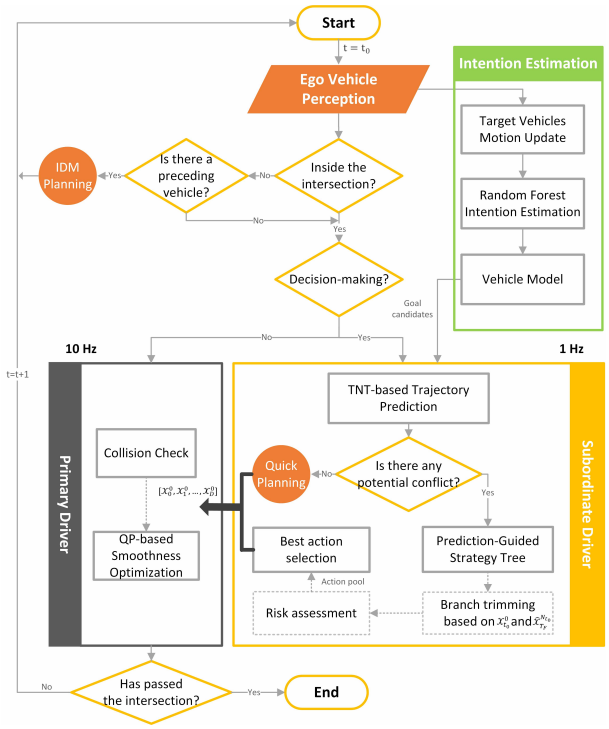


Fig. 2. The flow chart of the proposed framework, mainly involves the PD and SD running in different frequencies. PD conducts motion planning in real-time, while SD is responsible for decision-making. In SD, based on the prediction results from the neural network, the prediction-guided strategy tree is put into operation to get the optimal action sequence.

A. Observation Branch Trimming

The process of PGST is illustrated as Alg. 1. Given the perception messages, the TNT model takes the historical states of the vehicles $\mathcal{X}_{T_H}$, the predicted intentions $\zeta_{t_0}^{N_{t_0}}$, the vector of center lines L_m, and the sampled goals $\mathcal{G}_{tar}^{N_{t_0}}$ as input, and generates a bunch of k_{TNT} trajectories $\hat{\mathcal{X}}_{T_F}^n = \{\hat{\mathcal{X}}_{T_F}^{n,1} \cdots, \hat{\mathcal{X}}_{T_F}^{n,k_{\text{TNT}}}\}$ as well as their scores $\hat{\mathcal{Q}}_{T_F}^n = \{\hat{q}^{n,1} \cdots, \hat{q}^{n,k_{\text{TNT}}}\}$ for each target vehicle n. In POMDP, forward simulations are conducted in a series of sampled scenarios, which is the permutation and combination of the observations from the belief, causing the dimension explosion. In our algorithm, all the predicted trajectories are considered concurrently with different attention according to the predicted scores. By this means, most of the possible and dangerous situations are included given the accurate prediction, and all the raw actions could be evaluated under the same situation with an explicit risk assessment.

B. Action Branch Trimming

At each step d $\in D$ of the expansion, the feasible action pairs will be generated as the nodes, leading to the state sequence $\mathcal{X}_T$ in future $T = 1$s. The actions will be evaluated with security metrics and selected as parent nodes to further expand by the sorted scores.

Different from the fixed action samples in many works [20], [21], we generate adaptive action set $\boldsymbol{A}_{\text{raw}}$ containing action pairs of acceleration set and semantic path candidates. Given the current acceleration a_0, we consider the

Algorithm 1 Prediction-Guided Strategy Tree Search

Input: Number of social vehicles: N_{t_0}, States: $\mathcal{X}_{T_H}$, Intentions: $\zeta_{t_0}^{N_{t_0}}$, Goal candidates: $\mathcal{G}_{tar}^{N_{t_0}}$ Center-lines: L_m, Path candidates of ego vehicle: l_m^0, Path id of ego vehicle: $l_{t_0}^0$

Output: Optimal action sequence: $\mathbf{a}_{T_F}^* = \{(a_{\mathrm{d}}, \mathrm{id}_{\mathrm{d}}) | \mathrm{d} = 1, \cdots, D\}$

$\hat{\mathcal{X}}_{T_F}^{N_{t_0}}, \hat{\mathcal{Q}}_{T_F}^{N_{t_0}}$ = TNTPredict($\mathcal{X}_{T_H}, \zeta_{t_0}^{N_{t_0}}, L_m, \mathcal{G}_{tar}^{N_{t_0}}$)
$\boldsymbol{A}_{\text{safe}} \leftarrow \emptyset$, $\boldsymbol{A}_{\text{pool}} \leftarrow \emptyset$, $\boldsymbol{S}_{\text{pool}} \leftarrow \emptyset$
for $\mathrm{d} \in D$ **do**
  $\boldsymbol{A}_{\text{raw}}$=GenerateRawActions($\mathcal{X}_{t_0}^0$, $\boldsymbol{A}_{\text{safe}}$)
  for $\mathbf{a_i} \in \boldsymbol{A}_{\text{raw}}$ **do**
    $\mathcal{X}_T^0$ = ForwardSimulation($\mathcal{X}_{t_0}^0, \mathbf{a}_i, l_m^0, l_{t_0}^0$)
    G_T = GeneratePolygon($\mathcal{X}_T^0, \hat{\mathcal{X}}_T^{N_{t_0}}$)
    $\boldsymbol{A}_{\text{safe}}, \boldsymbol{A}_{\text{pool}}, \boldsymbol{S}_{\text{pool}}$=GenerateActionBranch($\mathcal{X}_T^0, \hat{\mathcal{X}}_T^{N_{t_0}}, G_T, l_m^0$)
    $\boldsymbol{A}_{\text{safe}}, \boldsymbol{S}_{\text{pool}}$=Reselect($\boldsymbol{A}_{\text{safe}}, \boldsymbol{A}_{\text{pool}}, \boldsymbol{S}_{\text{pool}}$)
  end for
end for
return $\mathbf{a}_{T_F}^*$=ScoringAndSelect($\boldsymbol{A}_{\text{safe}}, \boldsymbol{S}_{\text{pool}}$)

max jerk to get an action set $\mathbf{A} = \{\max(a_{\min}, a_0 - jerk), a_0, \min(a_{\max}, a_0 + jerk)\}$, in which the values are constrained by the limits. In lateral direction of Frenet frame, adjacent lanes in the candidate paths l_m^0 will be selected to form the path set $\mathbf{I} = \{\max(\mathrm{id}_{\min}, \mathrm{id}-1), \mathrm{id}, \min(\mathrm{id}_{\max}, \mathrm{id}+1)\}$. Raw action pairs are the combination of the acceleration and path candidates, as $\boldsymbol{A}_{\text{raw}} = \mathbf{A} \otimes \mathbf{I}, (a_j, \mathrm{id}_j) \in \boldsymbol{A}_{\text{raw}}$. For the safety purpose, it is permitted to change the path only once during the search depth $D = 3$. Then, the forward simulation will be performed using each action pair to generate the future trajectories $\mathcal{X}_T^0$ of the ego vehicle, which obeys the kinematic bicycle model [22] shown as (2) with discrete sampling time Δt, half of the length backward l_r and forward l_f from the mass center. δ_{f_t} is the front wheel steering angle and β_t is the angle between the velocity and the vehicle's direction of travel.

$$\begin{aligned} x_{t+1} &= x_t + v_t \cos(\phi_t + \beta_t)\Delta t, \\ y_{t+1} &= y_t + v_t \sin(\phi_t + \beta_t)\Delta t, \\ \phi_{t+1} &= \phi_t + v_t / l_r \sin(\beta_t)\Delta t, \\ v_{t+1} &= v_t + a_j \Delta t, \\ \beta_t &= \arctan(l_r/(l_r + l_f)\tan(\delta_{f_t})). \end{aligned} \tag{2}$$

Nodes of the action candidates are evaluated the risk assessment (security metrics). To increase the diversity of the actions and also avoid the overly conservative actions, the lists $\boldsymbol{A}_{\text{safe}}$ and $\boldsymbol{A}_{\text{pool}}$ are built. Completely safe actions with zero risk will be added to the $\boldsymbol{A}_{\text{safe}}$ directly, while other actions and the risk scores will be stored in $\boldsymbol{A}_{\text{pool}}$ and $\boldsymbol{S}_{\text{pool}}$ temporarily. If the amount $|\boldsymbol{A}_{\text{safe}}|$ exceeds the threshold $N_{th} = 5$, risky actions are discarded, otherwise, actions from $\boldsymbol{A}_{\text{pool}}$ are reselected by the sorted results in $\boldsymbol{S}_{\text{pool}}$. In this way, even the risky actions may be selected to continue the exploration, making it possible to plan out a decision with quantified security scores in the congested situation.

The risk score of each action pair is calculated based on the prediction results. For each action pair, a list consisted of $|k_{\text{TNT}} N_{t_0}|$ elements is kept, where each element represents the conflicts between the roll-out trajectory of the ego vehicle $\mathcal{X}_T^0$ and the k_{th} predicted trajectory of vehicle n, and the maximum risk score is regarded as the final risk assessment. To check the potential conflicts, we generate polygons G_T for the trajectories and check their intersections. The boundaries of the polygons are outlined by the shape of the vehicle plus a safety constant [23]. No intersections between polygons indicate no collision in T, and action a_j with risk factor $r_{\text{risk}} = 0.0$ will be considered completely safe. Otherwise, the potential collision will be regarded as a hidden risk and the risk factor r_{risk} will be calculated as (3), where $\hat{q}^{n,k} \in \hat{\mathcal{Q}}_{T_F}^n$ and t_c is the potential collision time ($t_c^{n,k}$ means the timestamp when vehicle n enters the collision zone indicated by the k_{th} prediction results). The risk factor becomes higher when the collision time is close and the prediction score is high. Furthermore, we plan to have the prediction bias included in (3) to consider the uncertainty increasing over time.

$$r_{\text{risk}} = \max_n(\max_{k \in k_{\text{TNT}}}(\hat{q}^{n,k} \exp(-|t_c^0 - t_c^{n,k}|))). \tag{3}$$

C. Penalty Function

For each action sequence $\mathbf{a}_{T_F} = \{\mathbf{a}_0, \cdots, \mathbf{a}_D\}$ in action sequences $\boldsymbol{A}_{\text{safe}}$, we could get the corresponding states $\mathcal{X}_{T_F}^0 = \{\mathcal{X}_0^0, \cdots, \mathcal{X}_D^0\}$ by vehicle model with risk factors. We consider the penalty of each action sequence and pick out $\mathbf{a}_{T_F}^*$ with minimum penalty as the strategy. The overall penalty $\mathcal{F}_{\text{all}}$ contains terms of safety, comfort, candidate path change, the deviation to the desired velocity, and the distance to the goal, which is formulated as a linear combination of the coefficients λ and the user-defined metrics in (4).

$$\begin{aligned} \mathcal{F}_{\text{all}} = \lambda_{\text{safe}}\mathcal{F}_{\text{safe}} + \lambda_{\text{com}}\mathcal{F}_{\text{com}} + \\ \lambda_{\text{dev}}\mathcal{F}_{\text{dev}} + \lambda_{\text{des}}\mathcal{F}_{\text{des}} + \lambda_{\text{goal}}\mathcal{F}_{\text{goal}}. \end{aligned} \tag{4}$$

Safety is the critically important factor in the evaluation metrics, which represents the uncertainty of collision between the ego vehicle and the social vehicles in (5). Completely safe action with $r_{\text{risk}} = 0.0$ results in no penalty and actions with $r_{\text{risk}} < 1$ will be penalized. $\mathbf{F}_{\text{safe}}$ is the constant for safety evaluation (the same below).

$$\mathcal{F}_{\text{safe}} = \mathbf{F}_{\text{safe}}(\sum_{\mathrm{d}=1}^{D} \gamma^{\mathrm{d}-1} \mathcal{X}_{\mathrm{d}}^0.r_{\text{risk}}). \tag{5}$$

As for comfort, the acceleration and also the jerk are taken into consideration, as in (6).

$$\mathcal{F}_{\text{com}} = \mathbf{F}_{\text{com}}(\sum_{\mathrm{d}=1}^{D} \gamma^{\mathrm{d}-1} (\mathcal{X}_{\mathrm{d}}^0.a + \mathcal{X}_{\mathrm{d}}^0.jerk)). \tag{6}$$

The penalty for deviation of a path refers to the path change cost. Since in intersections, it could be risky to turn

the steering wheel back and forth, penalties will be given if lane-change behavior occurs as in (7).

$$\mathcal{F}_{\text{dev}} = \mathbf{F}_{\text{dev}}(\sum_{\text{d}=1}^{D} \gamma^{\text{d}-1}(\mathcal{X}_{\text{d}}^{0}.\text{id} - \mathcal{X}_{\text{d}-1}^{0}.\text{id})). \tag{7}$$

The penalty for the difference between the actual velocity and the desired velocity is shown in (8) and the penalty for destination is shown in (9), where $\mathcal{X}_{\text{goal}}$ is the expected exit coordinates.

$$\mathcal{F}_{\text{des}} = \mathbf{F}_{\text{des}}(\sum_{\text{d}=1}^{D} \gamma^{\text{d}-1}(\mathcal{X}_{\text{d}}^{0}.v - v_{\text{des}})). \tag{8}$$

$$\mathcal{F}_{\text{goal}} = \mathbf{F}_{\text{goal}}[(\mathcal{X}_{D}^{0}.x - \mathcal{X}_{\text{goal}}.x)^2 + (\mathcal{X}_{D}^{0}.y - \mathcal{X}_{\text{goal}}.y)^2]. \tag{9}$$

γ is the discount factor to deal with the uncertainty caused by time lapse.

As the process in Fig. 2, after obtaining the optimal action $\mathbf{a}_{T_F}^{*}$ and the planned states $\mathcal{X}_{T_F}^{0}$, PD will conduct collision check and trajectory optimization to get a safe and smooth trajectory, which can be referred to IV. C in [24].

III. SIMULATION RESULTS

We validate the effectiveness of the proposed prediction-guided strategy tree search algorithm conditioned on the work of the neural network. To recreate the live traffic scene, we choose the data INTERACTION [25] collected from the real world, which contains naturalistic motions of various traffic participants in a variety of highly interactive driving scenarios. Since in real-world, it is hard to predict which of the two negotiating entities will make compromises, we make the following assumptions: when there is no dangerous conflict between the social vehicle and the ego vehicle, the social vehicle navigates according to the recorded trajectory from the dataset, which reflects the realistic driving characteristics of human drivers. When there exists a potential collision, it is stochastically decided which entity to yield or cross, to coincide with the real traffic situation and increase the diversity of simulation.

We first preprocessed the INTERACTION data and selected a total of 139 vehicles crossing the intersection, including 93 vehicles going straight, 34 vehicles turning left, and 13 vehicles turning right. We regard each vehicle as the ego vehicle and plan its trajectory from approaching the intersection to the exit.

In Fig. 3, for the comparison between the ground truth and the planned results by the proposed algorithm, we denote the driver-driven ego vehicle as a red rectangle and the corresponding ground truth as the red curve. The ego vehicle piloted by our method is represented by a green rectangle marked 'ego', and the planned trajectory in each decision round is the curve, whose speed profile is measured by a color bar. The candidate trajectories aligned with the road geometry are yellow dotted lines. The social vehicles are described as blue rectangles, and the rainbow colors represent the prediction results generated by the neural network, with red to purple representing the ranking scores from high to low.

A. Cases with Different Intentions

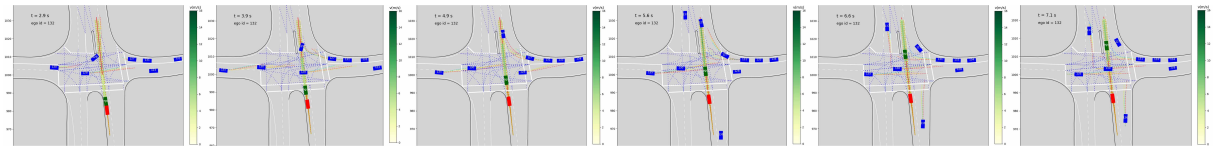
Fig. 3 shows representative cases with different intentions, including vehicle (ID: 132) going straight, vehicle (ID:88) turning right, and vehicle (ID: 144) turning left. In Fig. 4, sub-figure 1 depicts the distance, and sub-figure 2 shows the velocity along with the timestamp of decision-making. Sub-figure 3 shows the acceleration, with red and green rectangles representing the interaction process by deceleration and acceleration.

When the ego vehicle (ID: 132) is approaching the intersection, the social vehicle 128 shares interaction with it, as in Fig. 3 a). By the neural network, the ego vehicle successfully predicts the possible danger and decelerates to a low-speed trajectory. At around 4.7 s, the ego vehicle begins to interact with vehicle 130 and decides to accelerates to pass the conflict zone first. In consequence, the proposed method can help the vehicle understand the situation better and make a more appropriate decision. For human drivers, due to the lack of accurate predictions about other vehicles, overly conservative behaviors will cause unnecessary waiting time.

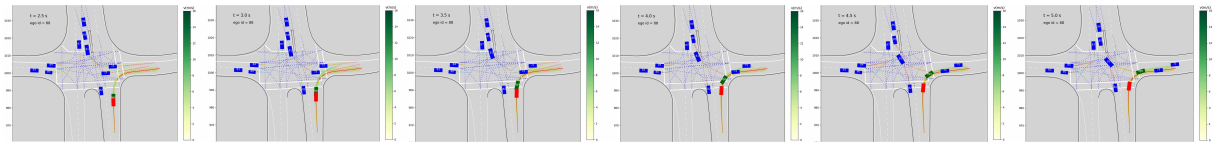
From the perspective of the right-of-way, right-turning vehicles need to yield to straight-going vehicles and left-turning vehicles (only motor vehicles are considered). As seen from Fig. 3 b), when the ego vehicle starts to turn right, the social vehicle 73 is going straight. Since vehicle 73 is closer to the exit of the intersection and has the right-of-way, the ego vehicle decides to give way to vehicle 73 by deceleration. In Fig. 4 b), after confirming that vehicle 73 has passed, the ego vehicle completes the right turn with almost constant speed.

Turning left is relatively complicated. For one reason, the tracks are not fixed to a regular trace. For another reason, the distance of the left turn is longer, which increases the probability of interaction even collision. When the ego vehicle (ID: 144) starts to turn left, vehicle 131 is found to go straight on the left side, so the ego vehicle chooses to slow down and gives its way. The constant deceleration before 4 s makes it successfully avoid vehicle 131. Finally, the ego vehicle accelerates slightly to pass through the intersection.

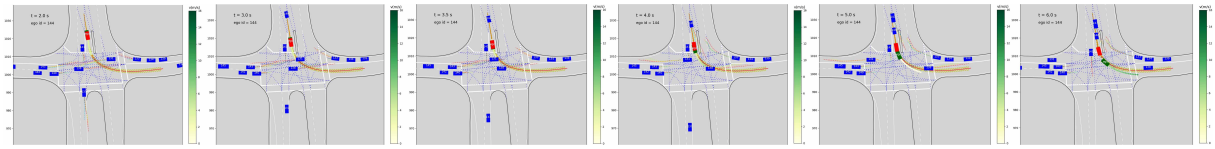
In many cases, when the traffic is dense, it will be a greater challenge for vehicles to pass through the intersection. Fig .3 d) shows the ego vehicle (ID: 124) intending to turn left. First, when approaching the intersection, the ego vehicle changes to the left-turn lane. The ego vehicle first estimates the straight-going behavior of vehicle 121, then decelerates to give way. The speed profile in Fig. 4 d) shows that before vehicle 121 passes the conflict area, the ego vehicle has been adopting the strategy of deceleration to avoid conflict. When it encounters vehicles 122 and 120, the negotiation of which one to yield or cross begins. For diversity, we make random choices to determine their reactions. One situation ends up with yielding the social vehicles, and the other leads to the acceleration strategy of the ego vehicle. The planned and yield curves in Fig. 4 d) show the different reactions of the ego vehicle.



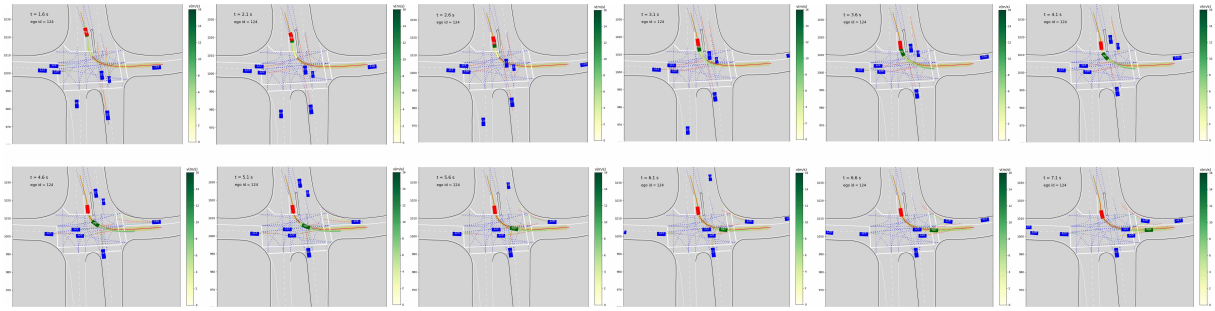
(a) Straight-going scenarios



(b) Right-turning scenarios



(c) Left-turning scenarios



(d) Complex left-turning scenarios

Fig. 3. Key frames in situations with different intentions.

B. Performance Analysis

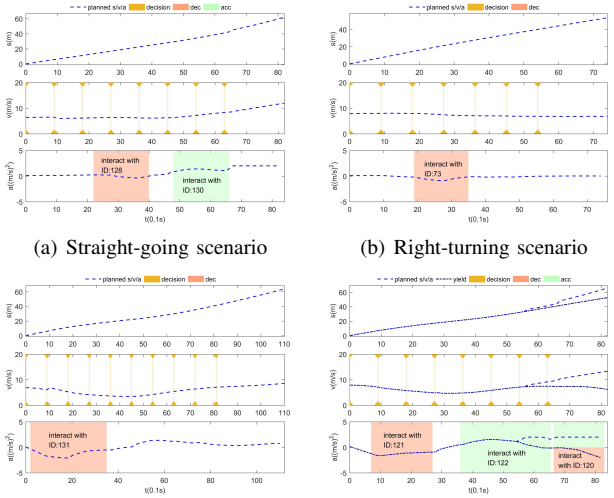
It is worth mentioning that the performance of the neural network keeps stable in the whole process. For example, in the straight-going scenario, the future trajectories are well predicted, of which social vehicle 129 turns left, vehicle 128 goes straight, and vehicle 127 turns right. Similarly, in the right-turn scenario, for decelerating vehicles 80 and 84, the neural network deduces their almost static states in the future. For vehicles 71 and 73 that are about to exit the intersection, it is predicted that they will accelerate according to the increasing speed. Because the prediction model utilizes the vectors of the road geometry, the predicted trajectories are well constrained within the boundaries. Moreover, since the graph neural network excels at depicting the mutual interaction of the vehicles, the predicted results are compatible with each other.

To compare the performance of the proposed algorithm and the human drivers, we calculate metrics in the aspects of travel smoothness, efficiency, and also risk ratio. For smoothness, we compute the variance of the velocity σ_v during the entire process. The greater variance means the unexpected stop-and-go behavior of the vehicle. The efficiency is quantified by the travel time T_{tr}, from the moment the ego vehicle begins to approach the entrance till it leaves the exit. To describe the risk ratio, we count the number of a re-planning process N_{re} when detecting a potential risk by checking the intersections between the inflated rectangles of vehicles [23]. The results are shown in Table I (GT means the ground truth), where we find that our method will lead to the steadily moving step and less time cost. The re-planning procedure will help to eliminate the risk, which can be controlled by adjusting the safe distance and parameters of the penalty function. The obvious difference is that human driver behaves overly conservatively. Even if it is available to drive into the intersection, they tend to wait and see, which even causes the 'dead lock'. Whereas our method takes the advantage of the prediction network to make more accurate predictions about the traffic situation and make appropriate strategies based on the evaluation parameters.

Among all the 139 tested vehicles, we have made statistics on the success rate, of which the success rate of going straight is 95.7%, turning left is 88.2%, and turning right is 92.3%. Left turns are with a low success rate due to the challenges aforementioned. After analyzing the failure cases, we summarize the reasons to three points. The visual blind spots, the prediction bias, and the trajectory optimization with strict constraints resulting in no solution.

IV. CONCLUSION

For the uncontrolled intersections with interaction and uncertainty, we propose a framework composed of a 'Primary



(a) Straight-going scenario (b) Right-turning scenario

(c) Left-turning scenario (d) Complex left-turning scenario

Fig. 4. Speed and acceleration profiles in situations with different intentions.

TABLE I
COMPARISON RESULTS

INTENTION	STRAIGHT		RIGHT		LEFT	
ALGORITHM	GT	PGST	GT	PGST	GT	PGST
$\sigma_v((m/s)^2)$	16.15	0.29	4.58	0.17	12.94	1.49
$T_{tr}(s)$	16.0	6.3	11.2	4.5	11.7	8.1
N_{re}	-	2	-	1	-	0

Driver (PD)' for planning and a 'Subordinate Driver (SD)' to make strategy. In SD, the 'prediction-guided strategy tree' is built on the learning-based prediction model to reduce the complexity of the strategy search process and realize risk assessment. This algorithm is used to search the optimal action sequence by trimming branches in both observation and action space. The experiments in cases with different driving intentions demonstrate that the proposed method results in more efficient and smooth trajectories than human-driven vehicles, which also consider the interaction of vehicles and the uncertainties that arise from the prediction. In the future, we plan to improve the performance of the prediction network and try to develop a closed-loop framework that can be used in the real-world.

REFERENCES

[1] N. Administration, "Fatality analysis reporting system," 2018, https://www-fars.nhtsa.dot.gov/ Accessed January 21, 2022.

[2] J. Pan, H. Sun, K. Xu, Y. Jiang, X. Xiao, J. Hu, and J. Miao, "Lane-attention: Predicting vehicles' moving trajectories by learning their attention over lanes," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7949–7956.

[3] R. Chandra, T. Guan, S. Panuganti, T. Mittal, U. Bhattacharya, A. Bera, and D. Manocha, "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4882–4890, 2020.

[4] X. Li, X. Ying, and M. C. Chuah, "Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving," *arXiv preprint arXiv:1907.07792*, 2020.

[5] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[6] 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.

[7] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, "Tnt: Target-driven trajectory prediction," *arXiv preprint arXiv:2008.08294*, 2020.

[8] P. Schörner, L. Töttel, J. Doll, and J. M. Zöllner, "Predictive trajectory planning in situations with hidden road users using partially observable markov decision processes," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2299–2306.

[9] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller, "A pomdp maneuver planner for occlusions in urban scenarios," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2172–2179.

[10] C. Hubmann, *Belief state planning for autonomous driving: Planning with interaction, uncertain prediction and uncertain perception*. KIT Scientific Publishing, 2021, vol. 48.

[11] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "Mpdm: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1670–1677.

[12] L. Zhang, W. Ding, J. Chen, and S. Shen, "Efficient uncertainty-aware decision-making for automated driving using guided branching," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3291–3297.

[13] W. Ding, L. Zhang, J. Chen, and S. Shen, "Epsilon: An efficient planning system for automated vehicles in highly interactive environments," *IEEE Transactions on Robotics*, 2021.

[14] T. Tram, I. Batkovic, M. Ali, and J. Sjöberg, "Learning when to drive in intersections by combining reinforcement learning and model predictive control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3263–3268.

[15] J. Li, L. Sun, M. Tomizuka, and W. Zhan, "A safe hierarchical planning framework for complex driving scenarios based on reinforcement learning," *arXiv preprint arXiv:2101.06778*, 2021.

[16] Y. Lee, P. Cai, and D. Hsu, "Magic: Learning macro-actions for online pomdp planning using generator-critic," *arXiv preprint arXiv:2011.03813*, 2020.

[17] P. Cai, Y. Luo, A. Saxena, D. Hsu, and W. S. Lee, "Lets-drive: Driving in a crowd by learning from tree search," *arXiv preprint arXiv:1905.12197*, 2019.

[18] K. Liu, N. Li, H. E. Tseng, I. Kolmanovsky, and A. Girard, "Interaction-aware trajectory prediction and planning for autonomous vehicles in forced merge scenarios," *arXiv preprint arXiv:2112.07624*, 2021.

[19] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[20] C. Cheng, D. Yao, Y. Zhang, J. Li, and Y. Guo, "A vehicle passing model in non-signalized intersections based on non-cooperative game theory," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2286–2291.

[21] J. Li, L. Sun, W. Zhan, and M. Tomizuka, "Interaction-aware behavior planning for autonomous vehicles validated with real traffic data," in *Dynamic Systems and Control Conference*, vol. 84287. American Society of Mechanical Engineers, 2020, p. V002T31A005.

[22] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[23] G. S. Sankar and K. Han, "Adaptive robust game-theoretic decision making strategy for autonomous vehicles in highway," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 484–14 493, 2020.

[24] T. Zhang, W. Song, M. Fu, Y. Yang, X. Tian, and M. Wang, "A unified framework integrating decision making and trajectory planning based on spatio-temporal voxels for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2021.

[25] 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.