2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Efficient Uncertainty-aware Decision-making for Automated Driving Using Guided Branching

Lu Zhang*, Wenchao Ding*, Jing Chen, and Shaojie Shen

***Abstract*— Decision-making in dense traffic scenarios is challenging for automated vehicles (AVs) due to potentially stochastic behaviors of other traffic participants and perception uncertainties (e.g., tracking noise and prediction errors, etc.). Although the *partially observable Markov decision process* (POMDP) provides a systematic way to incorporate these uncertainties, it quickly becomes computationally intractable when scaled to the real-world large-size problem. In this paper, we present an efficient uncertainty-aware decision-making (EUDM) framework, which generates long-term lateral and longitudinal behaviors in complex driving environments in real-time. The computation complexity is controlled to an appropriate level by two novel techniques, namely, the *domain-specific closed-loop policy tree* (DCP-Tree) structure and *conditional focused branching* (CFB) mechanism. The key idea is utilizing domain-specific expert knowledge to guide the branching in both action and intention space. The proposed framework is validated using both onboard sensing data captured by a real vehicle and an interactive multi-agent simulation platform. We also release the code of our framework to accommodate benchmarking.**

I. Introduction

In recent years, work in the areas of multi-sensor perception, prediction, decision-making, trajectory planning and control has enabled automated driving in difficult environments with other traffic participants and obstacles. Reasoning about hidden intentions of other agents is the key capability for a safe and robust automated driving system. However, even given perfect perception, it is still challenging to make safe and efficient decisions due to uncertain and sometimes unpredictable intentions of other agents. The situation is even worse when considering other system uncertainties such as imperfect tracking results and prediction errors.

There has been extensive literature on decision-making under uncertainty. Partially observable Markov decision process (POMDP) [1] provides a general and principled mathematical framework for planning in partially observable stochastic environments. However, due to the *curse of dimensionality*, POMDP quickly becomes computationally intractable when the problem size scales [2].

To address the computation difficulties, online POMDP planning algorithms [3] interleave the planning and execution and only reason about in the neighborhood of the current belief. Online POMDP solvers such as POMCP [4], DESPOT [5, 6] and ABT [7] have been proposed. The latest advances in the POMDP solvers are applied to many uncertainty-aware planning algorithms for automated vehicles [8]–[13]. However, despite that various simplifications and discretizations are applied, the efficiency of the existing methods is still inadequate for highly dynamic driving scenarios (see Sec. II for a detailed study).

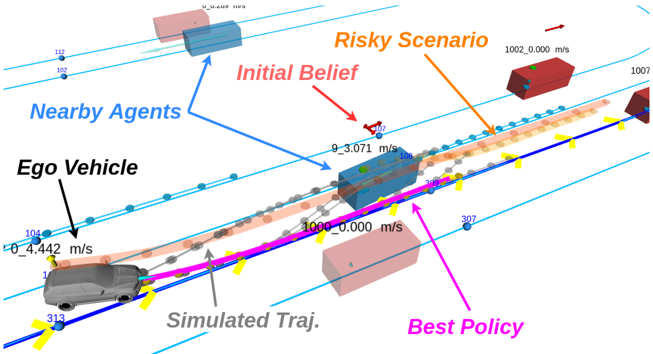


Fig. 1: Illustration of the proposed decision-making framework. The initial belief for different intentions (i.e., LC, LK) is marked by the *red* arrows on the agent vehicles with length representing the probability. The ego vehicle generates a set of sequential semantic-level policies according to the DCP-Tree. Each behavior sequence is simulated in closed-loop (marked by *black* dots) considering nearby agents. The risky scenario (in *orange*) in which the leading vehicle inserts to the ego target lane is identified by our CFB mechanism.

It is essential to incorporate domain knowledge to efficiently make robust decisions. Multipolicy decision-making (MPDM) [14]–[16] conducts deterministic closed-loop forward simulation of a finite discrete set of semantic-level policies (e.g., lane change (LC), lane keeping (LK), etc.) for the controlled (ego) vehicle and other agents, rather than performing the evaluation for every possible control input for every vehicle. However, the semantic behaviors for all the agents are assumed to be fixed in the whole planning horizon, which may not be true in long-term decision-making. Moreover, risk can be underestimated if initial behavior prediction is inaccurate [17, 18], which may lead to unsafe decisions.

Our goal here is to control the computational complexity of the decision-making problem to enable real-time execution, while retaining sufficient flexibility and fidelity to preserve safety. In this paper, we present an efficient uncertainty-aware decision-making (EUDM) framework. First, EUDM uses a *domain-specific closed-loop policy tree* (DCP-Tree) to construct a semantic-level action space. Each node in the policy tree is a finite-horizon semantic behavior of the ego

*These authors contributed equally to this work. All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. `lzhangbz@ust.hk`, `wdingae@ust.hk`, `jchenbr@ust.hk`, `eeshaojie@ust.hk` This work was supported in part by the HKUST-DJI Joint Innovation Laboratory and in part by the Hong Kong Ph.D. Fellowship Scheme.

vehicle. Each trace from the root node to the leaf node then represents a sequence of semantic actions of the ego vehicle. Each trace is evaluated in the form of closed-loop simulation similar to [14] but the ego behavior is allowed to change in the planning horizon.

The DCP-Tree essentially determines a preliminary semantic-level action sequence for the ego vehicle, however, the behaviors (intentions) of other agent vehicles remain undetermined. Since the combinations of the intentions of agent vehicles explode exponentially, it is inefficient to naively sample all possible combinations of agent intentions. To overcome this, EUDM uses the *conditional focused branching* (CFB) mechanism to pick out the potentially risky scenarios using open-loop safety assessment conditioning on the ego action sequence. EUDM is highly *parallelizable*, and can produce long-term (up to 8 s) lateral and longitudinal fine-grained behavior plans in real-time (20 Hz).

The major contributions are summarized as follows:

- An efficient uncertainty-aware decision-making framework for automated driving.
- A real-time and *open-source*[1] implementation of the proposed decision-making framework.
- Comprehensive experiments and comparisons are presented to validate the performance, using both onboard sensing data captured by a real vehicle and an interactive multi-agent simulation platform.

The remainder of this paper is organized as follows. The related work is reviewed in Section II. An overview of the proposed decision-making framework is provided in Section III. The methodology and implementation are detailed in Section IV and Section V, respectively. Experimental results and benchmark analysis are elaborated in Section VI. Finally, this paper is concluded in Section VII.

II. Related Work

There is extensive literature on decision-making for automated driving [19, 20]. Many researchers tackle the planning problem in a decoupled manner, namely, “predict and plan” [21]–[24]. Specifically, prediction results are fixed in one planning cycle. Several drawbacks may exist. First, it is problematic to handle interaction among agents in this decoupled design. Second, given onboard sensing, imperfect tracking may result in prediction errors, which affects the safety of the decision. Third, even given perfect perception, the prediction uncertainty still scales dramatically w.r.t. the prediction horizon [25] due to partial observability.

POMDP is a powerful tool to handle various uncertainties in the driving task using a general probabilistic framework [26]. However, due to the *curse of dimensionality*, POMDP quickly becomes computationally intractable when the problem size scales [2].

Leveraging the advance of online POMDP solvers [5]–[7, 27], several methods are proposed to tackle the decision-making problem by simplifying the system model and restricting the problem scale. Bai *et al.* [10] decouple the planning problem into pathfinding and velocity planning, and POMDP is only applied to the velocity planning. Hubmann *et al.* proposed POMDP-based decision-making methods for urban intersection [12] and merging [13] scenarios. However, human driving knowledge is not incorporated into the heuristic design. Meanwhile, the efficiency is still inadequate (less than 5 Hz) in fast-changing environments.

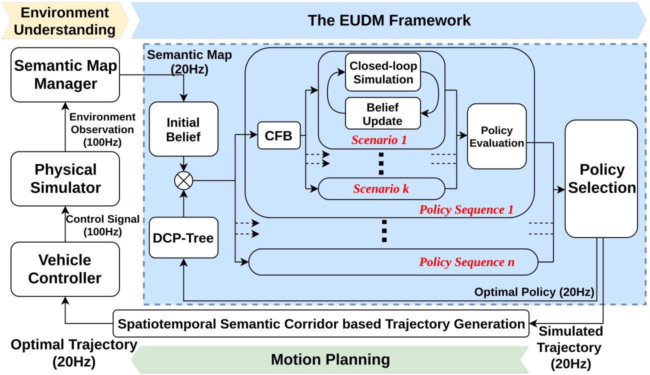


Fig. 2: Illustration of the proposed decision-making framework (in the *blue* box) and its relationship with other system components.

Cunninghan *et al.* [14, 16] proposed the multipolicy decision-making (MPDM) framework, which approximates the POMDP process into the closed-loop simulation of predefined semantic-level driving policies (e.g., LC, LK, etc.) for all the agents. The incorporation of domain knowledge greatly accelerates the problem-solving. However, the ego behavior is fixed for the whole planning horizon, which may result in reactive decisions. Moreover, the hidden intentions (driving policy of other agents) are sampled according to initial behavioral prediction (initial belief) and will not be updated during the simulation. As a result, risky outcomes may not be reflected in policy evaluation due to inaccurate initial behavior prediction or insufficient intention samples [17].

In this paper, we follow the idea of semantic-level closed-loop policies from MPDM. However, there are two major differences. First, the policy of the ego vehicle is allowed to change in the planning horizon according to the DCP-Tree, which makes it suitable for long-term decision-making. Second, focused branching is applied to pick out the risky scenarios, even given totally uncertain behavior prediction, which enhances the safety of the framework.

III. System Overview

An overview of our system is shown in Fig. 2, which is similar to our previous work [28]. The difference is that the focus of this paper is the decision-making part. Different from the methods which decouple the prediction and planning module [21]–[24], for our method, the intentions of agents are tracked and updated in the planning horizon.

In EUDM, DCP-Tree is used to guide the branching in the action domain and update the semantic-level policy tree based on the previous best policy. Each action sequence of the ego vehicle is then scheduled to a separate thread. For each ego action sequence, the CFB mechanism is applied to pick out risky hidden intentions of nearby vehicles and achieves guided branching in intention space. The output of

[1] https://github.com/HKUST-Aerial-Robotics/eudm_planner

the CFB process is a set of scenarios containing different hidden intentions combinations of nearby vehicles. Each scenario is then evaluated by the closed-loop simulation to account for interaction among agents in a sub-thread in parallel. All the scenarios are fed to the cost evaluation module and biased penalty is applied to risky branches. The output of the EUDM framework is the best policy which is represented by a series of discrete vehicle states (0.4 s resolution in the experiments) generated by the closed-loop forward simulation. The state sequence is fed to the motion planner to guide the trajectory generation process [28].

IV. Decision-making via Guided Branching

A. Preliminaries on POMDP

A POMDP can be defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O}, \gamma \rangle$, which are the state space, action space, state-transition function, reward function, observation space, observation function, and discount factor, respectively. The state of the agent is *partially observable*, and is described as a belief b, which is a probability distribution over $\mathcal{S}$. The belief state can be updated given an action a and an observation z using Bayes' inference $b_t = \tau(b_{t-1}, a_{t-1}, z_t)$. The goal of the online POMDP planner is finding an optimal policy π^* that maximizes the total expected discounted reward, given an initial belief state b_0 over the planning horizon t_h. We refer interested readers to [4, 26] for more details.

The optimal policy is often pursued using a multi-step look-ahead search starting from the current belief b_0. A belief tree can be expanded using the *belief update* function after taking actions and receiving observations during the search. However, the scale of the belief tree grows exponentially ($\mathcal{O}(|\mathcal{A}|^h|\mathcal{Z}|^h)$) with respect to the tree depth h, which is computationally intractable given large action space $|\mathcal{A}|$ and observation space $|\mathcal{Z}|$. State-of-the-art online POMDP planners [5, 7, 27] use Monte-Carlo sampling to deal with the *Curse of Dimensionality* and *Curse of History* [27]. Meanwhile, generic heuristic search such as branch-and-bound [5] and reachability analysis [7] can be used to accelerate the search. Note that the focus of this paper is to utilize domain-specific knowledge to achieve guided branching, which is also compatible with the generic heuristic search techniques.

B. Domain-specific Closed-loop Policy Tree

As pointed out by MPDM [15], for the decision-making problem, too much computation effort of POMDP is spent on exploring the space that is unlikely to be visited. The key feature of MPDM is using semantic-level policies instead of traditional "state"-level actions (e.g., discretized accelerations or velocities). By using semantic-level policies, the exploration of the state space is guided by simple closed-loop controllers (i.e., domain knowledge).

Motivated by MPDM, we also use semantic-level policies, as one source of the domain knowledge. However, as elaborated in Sec. II, one major limitation of MPDM is that the semantic-level policy of the ego vehicle is not allowed to change in the planning horizon. For example, MPDM may simulate LC and LK policies for the whole planning horizon (e.g., 8 s). Typical patterns such as lane change in different stages (e.g., in 2 s, 4 s, 6 s) are not included in its decision space. As a result, the decision of MPDM tends to be local and may not be suitable for long-term decision-making (see Fig. 3 for an example).

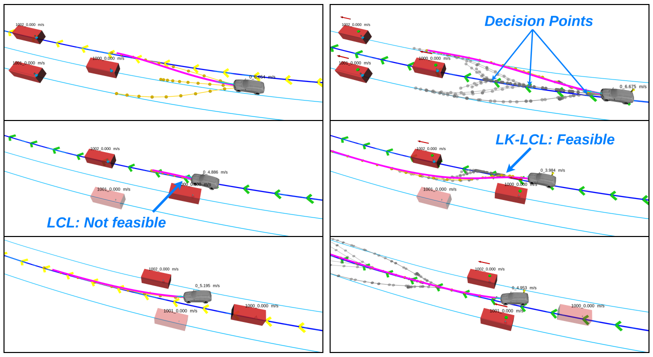


Fig. 3: Comparison of MPDM (left) and EUDM (right). The ego vehicle (*gray*) needs to conduct an overtaking maneuver. The simulated behaviors of MPDM are fixed in the planning horizon. The ego vehicle cannot make a lane-change-left (LCL) decision until it passes the blocking vehicle, so the generated plan is local and reactive. EUDM considers the change of behavior in different future stages, which results in a consistent and farsighted plan.

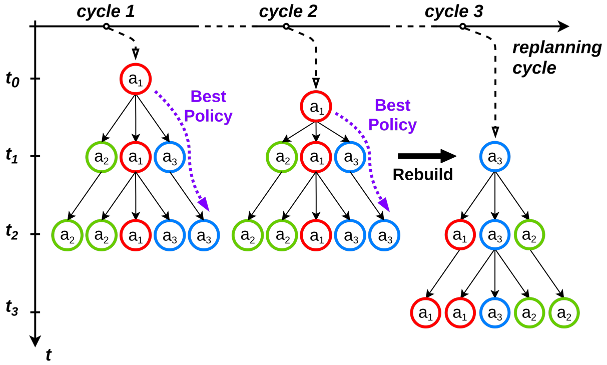


Fig. 4: Illustration of the proposed DCP-Tree and the rebuilding process after its *ongoing* action changes. Suppose there are three discrete semantic-level actions $\{a_1, a_2, a_3\}$ and the height of the tree is three. The *ongoing* action for the left and middle tree is a_1, and the best policy is the dashed *purple* trace. After executing a_1, the *ongoing* action switches to a_3, and the DCP-Tree is updated to the right tree. Each trace only contains one change of action.

In this paper, DCP-Tree is utilized to generate future ego action sequences, which allows the semantic policy of the ego vehicle to change in the planning horizon. The nodes of DCP-Tree are pre-defined semantic-level actions associated with a certain time duration. The directed edges of the tree represent the execution order in time. DCP-Tree origins from an *ongoing* action $\hat{a}$, which is the executing semantic-level action from the best policy in the last planning cycle. Every time we enter a new planning episode, the DCP-Tree is rebuilt by setting $\hat{a}$ to the root node.

Since the ego policy is allowed to change in the planning horizon, the challenge is that the number of possible policy sequences scales exponentially w.r.t. the depth of the tree (i.e., the planning horizon). To overcome this, DCP-Tree is

Algorithm 1: Process of EUDM

```
1  Inputs: Current states of ego and other vehicles $s$;
     *Ongoing action* $\hat{a}$; Pre-defined semantic action set
     $\mathcal{A}$; Planning horizon $t_h$;
2  $\mathfrak{R} \leftarrow \emptyset$; // set of rewards for each policy;
3  $\Psi \leftarrow \texttt{UpdateDCPTree}(\mathcal{A}, \hat{\mathtt{a}})$; // DCP-Tree $\Psi$;
4  $\hat{\Pi} \leftarrow \texttt{ExtractPolicySequences}(\Psi)$;
5  **foreach** $\pi \in \hat{\Pi}$ **do**
6      $\Gamma^{\pi} \leftarrow \emptyset$; // set of simulated trajectories;
7      $\Omega \leftarrow \texttt{CFB}(s, \pi)$; // set of critical scenarios;
8      **foreach** $\omega \in \Omega$ **do**
9          $\Gamma^{\pi} \leftarrow \Gamma^{\pi} \cup \texttt{SimulateForward}(\omega, \pi, t_h)$;
10     **end**
11     $\mathfrak{R} \leftarrow \mathfrak{R} \cup \texttt{EvaluatePolicy}(\pi, \Gamma^{\pi})$;
12 **end**
13 $\pi^*, \hat{a} \leftarrow \texttt{SelectPolicy}(\mathfrak{R})$;
```

expanded by a pre-defined strategy, which comes from the observation that, for human drivers, typically we do not frequently change the driving policy back and forth in a single decision cycle. For example, human drivers often evaluate whether it is feasible to conduct one policy change, e.g., switching from LK to LC in several seconds. This does not prevent human drivers from conducting complex maneuvers since consecutive decisions from different decision cycles can be combined. Motivated by this, from the *ongoing* action, each policy sequence will contain at most one change of action in one planning cycle, as shown in Fig. 4, while the back-and-forth behavior is achieved by replanning.

For instance, suppose the ongoing action is LK, the resulting policy sequences may include (LK-LC-LC-LC...), (LK-LK-LC-LC...) and (LK-LK-LK-LC...), etc. Note that the size of leaf nodes in DCP-Tree is $\mathcal{O}[(|A|-1)(h-2)+|A|]$, $\forall h > 1$, which grows linearly with respect to the tree height h. It is also notable that MPDM is only one branch of our DCP-Tree and DCP-Tree includes multiple future decision points. Compared to MPDM, DCP-Tree has much larger decision space resulting in more flexible maneuvers as shown in Fig. 3. Moreover, we also observe a significant improvement of decision consistency among consecutive planning cycles compared to MPDM.

C. Conditional Focused Branching

Essentially, DCP-Tree provides a guided branching mechanism in the action space. The remaining problem is to determine the semantic-level intentions of the nearby vehicles, namely, the branching in the intention space. The challenge here is that the combination of intentions of nearby vehicles scales exponentially w.r.t. the number of agents. In the case of MPDM, the intention of the nearby vehicles is fixed for the whole planning horizon, and the initial intention is sampled according to a behavior prediction algorithm. The limitation of MPDM is that, with a limited number of samples, influential risky outcomes may not be rolled out, especially when the initial intention prediction is inaccurate [17].

To overcome this, we propose the CFB mechanism. The goal here is to find the intentions of nearby vehicles which potentially lead to risky outcomes with as few branches as possible. The term "conditional" means conditioning on the ego policy sequence. The motivation comes from the observation that the attention of the human driver for nearby vehicles is biased differently when intending to conduct different maneuvers. For example, a driver will pay much more attention to the situation on the left lane rather than the right one when he intends to make an LCL. As a result, by conditioning on the ego policy sequence, we can pick out a set of relevant vehicles w.r.t. the ego future actions. The selection process is currently based on rule-based expert knowledge as detailed in Sec. V. We point out that learning-based attention mechanisms can also be incorporated and we leave it as an important future work.

By conditioning on the ego policy sequence, we obtain a subset of vehicles that need to be further examined. Instead of enumerating all the possible intentions for this subset of vehicles, we introduce a preliminary safety check to pick out the vehicles which we should pay special attention to. The preliminary safety assessment is conducted using open-loop forward simulation based on *multiple hypotheses*. For example, for the vehicle whose intention is uncertain, we anticipate what the situation will be if the vehicle is LC or LK, respectively. The anticipation is carried out using open-loop forward simulation under the intention hypothesis. The idea of using open-loop simulation is that by ignoring the interactions among agents, we check how the serious the situation will be if surrounding agents are completely uncooperative and does not react to the other agents. For the vehicles which do not pass the preliminary safety assessment, different scenarios are further examined in closed-loop forward simulation. And for the vehicles which pass the assessment, we use maximum a posteriori (MAP) from initial belief. As a result, the branching in intention space is guided to potentially risky scenarios. In practice, we find that the preliminary safety check can identify many dangerous cases despite its simple design.

The flow of EUDM is described in Algo.1. Evaluation for each policy sequence can be carried out in parallel (Line 5 to 11). Each critical scenario selected by CFB is examined by closed-loop forward simulation (Line 8 to 10) in parallel. Each policy is evaluated (Line 11) using the reward function detailed in Sec. V and the best policy is elected (Line 13).

V. Implementation Details

A. Semantic-level Actions

We consider both lateral and longitudinal actions to ensure the diversity of the driving policy. Similar to [16], we define the lateral actions as {*LK, LCL, LCR*}. For longitudinal action, we use {*accelerate, maintain speed, decelerate*}. Note that these longitudinal actions are not discretized control signals such as acceleration commands in [10, 12] but continuous desired velocity applied to the forward simulation model. Each semantic-level action is assigned with time duration of 2 s, while the closed-loop simulation is carried out with 0.4 s resolution to preserve the simulation fidelity. Note that the duration of the ongoing action is deducted by the replanning resolution (0.05 s) for each planning cycle. The depth of the DCP-Tree is set as 4, thereby we obtain a planning horizon up to 8 s.

B. Forward Simulation

The goal of the closed-loop simulation is to push the state of the multi-agent system forward while considering the potential interaction. The simulation model should achieve a good balance between simulation fidelity and inference efficiency. We adopt the *intelligent driving model* [29] and *pure pursuit controller* [30] as the longitudinal and lateral simulation models, respectively. Control noises are injected to reflect the stochastic property of driving behaviors.

C. Belief Update

The hidden intentions considered in this work include lateral behaviors, such as $\{LK, LCL, LCR\}$. The belief over these intentions of agent vehicles are updated during the forward simulation as shown in Fig. 2. In this work, we adopt a rule-based lightweight belief tracking module that takes a set of features and metrics including velocity difference, distance w.r.t. the leading and following agents on the current and neighboring lanes, responsibility-sensitive safety (RSS) [31] and lane-changing model [32] as input[2]. The belief tracker generates a probability distribution over the intentions (i.e., LK, LCL, LCR). The probability serves as the importance weight during policy evaluation. In the experiments, we find the rule-based belief tracker works well despite its simple structure. Currently, we are exploring using learning-based belief trackers for intention tracking [25] which will be incorporated into the EUDM framework.

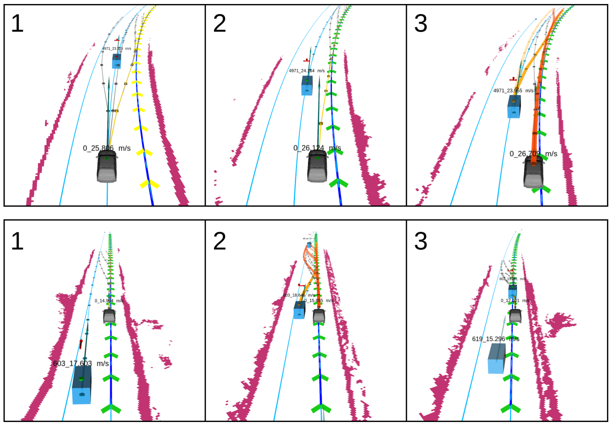


Fig. 5: Open-loop test using onboard sensing data. Up: The ego vehicle is approaching another vehicle and making a LC decision (1 and 2). After the ego vehicle finishing LCR, EUDM detects several risky scenarios due to the uncertain intention of the front vehicle. Bottom: Another vehicle overtakes the ego vehicle and merges into our lane aggressively (1 and 2). EUDM captures the risky situation and takes a proper deceleration policy (3).

D. CFB Mechanism

The first step of CFB is the key vehicle selection. For the current lane and neighboring lanes, we search forward and backward along the lanes for a certain distance w.r.t. the current speed of ego vehicle and the vehicles in this range are marked as key vehicles. The second step is uncertain vehicle selection according to the initial belief. Specifically, we pick out the vehicles, whose probabilities for the three intentions are close to each other, as uncertain vehicles. Note that for the vehicles with confident prediction, we select the MAP intention and marginalize the intention probabilities using the MAP selection result. The third step is using the open-loop forward simulation for safety assessment. For the vehicles which fail the assessment, we enumerate all the possible combinations of their intentions. Each combination becomes a CFB-selected scenario and the probability of scenario is calculated. The fourth step is picking out top k scenarios according to user-preference, and we further marginalize the probabilities among the top-k scenarios. The marginal probabilities become the weights of CFB-selected scenarios during evaluation.

E. Policy Evaluation

The overall reward for a policy sequence is calculated by the weighted summation of the reward for each CFB-selected scenario. The reward function consists of a linear combination of multiple user-defined metrics including efficiency (measured by the difference between current velocity and desired velocity), safety (measured by the distance between our vehicle and surrounding vehicles) and consistency (measured by the difference between the last best policy and the policy to be evaluated).

F. Trajectory Generation

The output of our behavior planner is a series of discrete states of the ego vehicle with 0.4 s resolution. The behavior plan is fed to the motion planner proposed in our previous work [28], which utilizes a spatio-temporal corridor structure to generate safe and dynamically feasible trajectories.

VI. Experimental Results

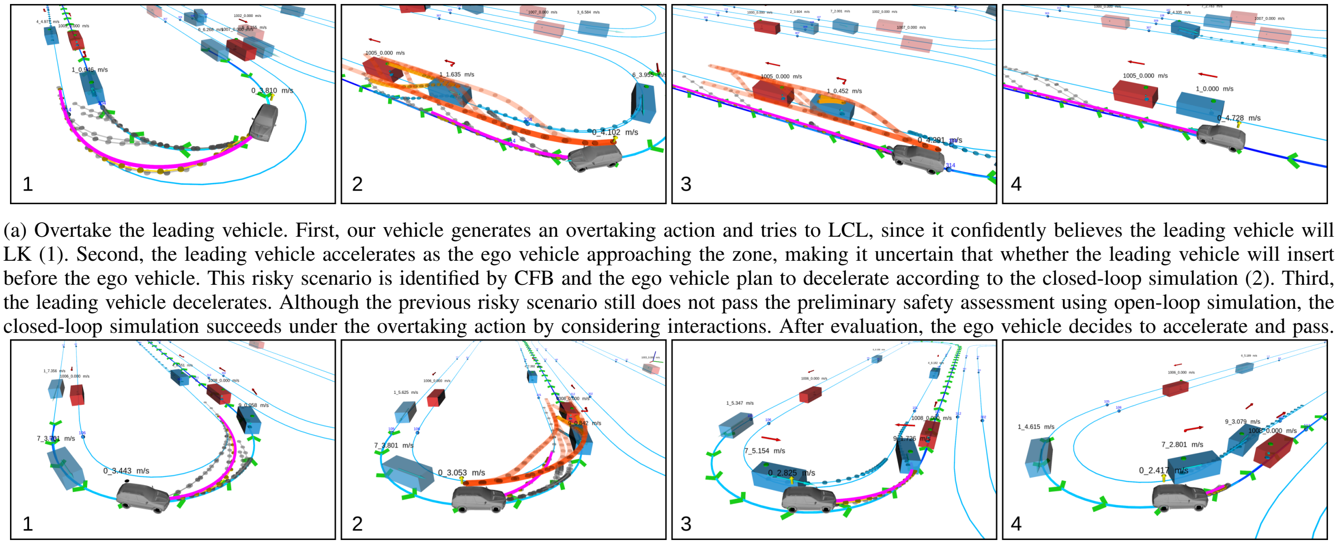
A. Simulation Platform and Environment

The experiment is conducted in an interactive multi-agent simulation platform as introduced in Sect.III. All agents can interact with each other without knowing the driving model of other vehicles. The forward simulation model used in the ego vehicle may differ from the actual model running on the agents. The proposed decision-making method is implemented in C++11. All the experiments are conducted on a desktop computer equipped with an Intel i7-8700K CPU, and our proposed method can run stably at 20 Hz.

B. Qualitative Results

To verify that our EUDM can generate flexible and consistent behaviors in highly interactive scenarios, we show several cases using both real-world data and simulation.

[2] Detailed implementation can be found in our open-source code.



(a) Overtake the leading vehicle. First, our vehicle generates an overtaking action and tries to LCL, since it confidently believes the leading vehicle will LK (1). Second, the leading vehicle accelerates as the ego vehicle approaching the zone, making it uncertain that whether the leading vehicle will insert before the ego vehicle. This risky scenario is identified by CFB and the ego vehicle plan to decelerate according to the closed-loop simulation (2). Third, the leading vehicle decelerates. Although the previous risky scenario still does not pass the preliminary safety assessment using open-loop simulation, the closed-loop simulation succeeds under the overtaking action by considering interactions. After evaluation, the ego vehicle decides to accelerate and pass.

(b) Give up overtaking and give way for other vehicles. First, our vehicle decides to LCL and overtake (1). Second, the leading vehicle accelerates, upon the assessment of potential risk, the ego vehicle decides to decelerate to follow the leading vehicle while conducting the LCL (2). Third, the leading vehicle conducts LCL and the following vehicle tries to overtake us (3). The ego vehicle then decides to give up LCL and yield (3 and 4).

Fig. 6: Illustration of different decision-making results in a conflict zone.

1) Overtaking and Yielding in a Conflict Zone: Conflict zones are common in urban traffic. As shown in Fig.6, the ego vehicle has to pass through a conflict zone where there is a leading vehicle also trying to LC and pass through. Moreover, the initial belief of the leading vehicle is uncertain given the current observation. As shown in Fig.6, the EUDM framework can automatically select appropriate behaviors (i.e., overtaking or yielding) depending on the situation.

TABLE I: Comparison of different decision-making approaches.

Map	**Method**	**Safety**	**Efficiency**	**Comfort** ($\#/km$)	
			Ave. Vel. (m/s)	UD	LCC
Double Merge	MPDM	0.048	4.9	1.85	4.63
	EDM	0.043	**5.3**	2.50	3.21
	EUDM	**0.025**	4.9	**0.38**	**1.53**
Ring	MPDM	0.042	13.36	1.09	0.0
	EDM	0.030	**14.37**	1.41	0.0
	EUDM	**0.003**	12.86	**0.48**	0.0

2) Testing using Real-world Onboard Sensing Data: In this case, our method is tested in an open-loop manner by using the data collected by a real automated vehicle. The goal is to verify whether the proposed method can capture risky scenarios and work under uncertain predictions and noisy perception. As shown in Fig. 5, EUDM can make appropriate decisions to overtake or decelerate depending on the situation.

C. Quantitative Results

We conduct a comprehensive quantitative comparison with the MPDM [16], which is one of the state-of-the-art decision-making methods for automated driving. We evaluate the two methods using two benchmark tracks, i.e., *Double Merge* and *Ring*. Detailed experiments can be found in the attached video, while the statistical results are shown in Table. I.

1) Metrics: We introduce three major metrics to evaluate the performance of two methods, namely, safety, efficiency, and comfort. For safety, we count the fraction of frames that the distance between ego vehicle and other agents smaller than a threshold (i.e., safety distance). The efficiency is represented by the average velocity of the ego vehicle. The comfort is described by the number of *uncomfortable deceleration* (UD) and the number of *large curvature changing* (LCC) per kilometers. The threshold of UD and LCC is set to 1.6 m/s^2 and 0.12 $(s \cdot m)^{-1}$, respectively.

2) Benchmarking: We also conduct an ablative study by removing the CFB mechanism from EUDM, which results in the EDM method as shown in Table. I. The goal is to verify whether the CFB can improve the robustness and safety of the framework. As shown in Table. I, EUDM makes safer decisions than EDM and MPDM according to the safety metric. The reason is that EUDM explicitly explores the risky scenarios and conducts biased branching. Note that the double merge map is a dense interactive scenario where unsafe situations are hard to be completely avoided due to aggressive behaviors of agent vehicles. For efficiency, our method and MPDM perform similarly, while EDM has a higher average velocity. It is because EDM enlarges the action space compared to MPDM, and it may take over-aggressive risky actions (see Fig.3). In terms of comfort, EUDM can generate much smoother behaviors than the other two baselines, since it takes conservative policy under risky scenarios beforehand and avoids hard brakes.

VII. Conclusion and Future Work

In this paper, we proposed the EUDM framework for automated driving in dense interactive scenarios, by introducing two novel techniques, namely, the DCP-Tree and CFB mechanism. The complete framework is open-sourced and comprehensive evaluations are conducted using both real-world data and simulation. In the future, we will conduct closed-loop field test for the EUDM framework.

REFERENCES

[1] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable markov processes over a finite horizon," *Operations research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[2] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems," in *AAAI/IAAI*, 1999, pp. 541–548.

[3] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, "Online planning algorithms for pomdps," *Journal of Artificial Intelligence Research*, vol. 32, pp. 663–704, 2008.

[4] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in neural information processing systems*, 2010, pp. 2164–2172.

[5] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "Despot: Online pomdp planning with regularization," *Journal of Artificial Intelligence Research*, vol. 58, pp. 231–266, 2017.

[6] P. Cai, Y. Luo, D. Hsu, and W. S. Lee, "Hyp-despot: A hybrid parallel algorithm for online planning under uncertainty," *Proc. of Robot.: Sci. and Syst.*, 2018.

[7] H. Kurniawati and V. Yadav, "An online pomdp solver for uncertainty planning in dynamic environment," in *Robotics Research*. Springer, 2016, pp. 611–629.

[8] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A pomdp approach," *Intl. J. Robot. Research*, vol. 33, no. 9, pp. 1288–1302, 2014.

[9] W. Liu, S.-W. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online pomdp," in *IEEE Intl. Veh. Sym.* IEEE, 2015, pp. 1126–1133.

[10] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* IEEE, 2015, pp. 454–460.

[11] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps," in *Proc. of the Intl. Conf. on Intel. Trans. Syst.* IEEE, 2014, pp. 392–399.

[12] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 5–17, 2018.

[13] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A belief state planner for interactive merge maneuvers in congested traffic," in *Proc. of the Intl. Conf. on Intel. Trans. Syst.* IEEE, 2018, pp. 1617–1624.

[14] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* IEEE, 2015.

[15] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction." in *Proc. of Robot.: Sci. and Syst.*, 2015.

[16] ——, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," *Autonomous Robots*, 2017.

[17] D. Mehta, G. Ferrer, and E. Olson, "Fast discovery of influential outcomes for risk-aware mpdm," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* IEEE, 2017, pp. 6210–6216.

[18] ——, "Backprop-mpdm: Faster risk-aware policy evaluation through efficient gradient optimization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1740–1746.

[19] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[20] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[21] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Intl. J. Robot. Research*, 2012.

[22] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* IEEE, 2011.

[23] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[24] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *IEEE Intl. Veh. Sym.* IEEE, 2014, pp. 458–464.

[25] W. Ding, J. Chen, and S. Shen, "Predicting vehicle behaviors over an extended horizon using behavior interaction network," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.* IEEE, 2019.

[26] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[27] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in neural information processing systems*, 2010, pp. 2164–2172.

[28] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2997–3004, 2019.

[29] 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.

[30] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.

[31] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[32] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.