

MARC: Multipolicy and Risk-Aware Contingency Planning for Autonomous Driving

Tong Li , Graduate Student Member, IEEE, Lu Zhang , Sikang Liu , and Shaojie Shen 

Abstract—Generating safe and non-conservative behaviors in dense, dynamic environments remains challenging for automated vehicles due to the stochastic nature of traffic participants’ behaviors and their implicit interaction with the ego vehicle. This letter presents a novel planning framework, Multipolicy And Risk-aware Contingency (MARC) planning, that systematically addresses these challenges by enhancing the multipolicy-based pipelines from both behavior and motion planning aspects. Specifically, MARC realizes a critical scenario set that reflects multiple possible futures conditioned on each semantic-level ego policy. Then, the generated policy-conditioned scenarios are further formulated into a tree-structured representation with a dynamic branchpoint based on the scene-level divergence. Moreover, to generate diverse driving maneuvers, we introduce risk-aware contingency planning, a bi-level optimization algorithm that simultaneously considers multiple future scenarios and user-defined risk tolerance levels. Owing to the more unified combination of behavior and motion planning layers, our framework achieves efficient decision-making and human-like driving maneuvers. Comprehensive experimental results demonstrate superior performance to other strong baselines in various environments.

Index Terms—Autonomous vehicle navigation, intelligent transportation systems, motion and path planning.

I. INTRODUCTION

OVER the past decade, autonomous driving has experienced remarkable progress. Nonetheless, ensuring safe and efficient operations in highly interactive environments remains a formidable challenge. The difficulties of planning under uncertainty result from imperfect observation and, more importantly, the inherently multimodal intentions of various road users that cannot be observed directly. A false estimation of the other’s purpose can lead the autonomous vehicle to generate overly cautious or hazardous driving behavior, putting traffic safety at risk.

The academic community has extensively explored methods to address the challenges. For interactive decision-making, existing methods [1], [2], [3], [4] usually utilize Markov

decision process (MDP) and partially observable Markov decision process (POMDP), mathematically rigorous formulations for decision processes in stochastic environments. However, solving such problems becomes computationally intractable as the size increases. To address this issue, multipolicy-based pipelines [5], [6], [7] are proposed, which prune the belief trees heavily and decompose the original problem into a limited number of closed-loop policy evaluations. These methods leverage domain-specific semantic-level policies (e.g., lane-keeping and lane-changing) to approximate the action space and realize multiple future scenarios for each policy to handle interaction uncertainties. However, these methods are typically designed to generate the *best* policy over all possible future evolutions, while the subsequent trajectory generation modules only account for a single selected scenario, making the whole system hard to exploit the multi-modality of decision-making.

Another way of interaction handling is generating contingency plans for multiple possible futures in the motion planning layers. Unlike the aforementioned decision-making algorithms, contingency planning [8], [9] is always formulated as a numerical optimization problem, in which the environmental uncertainties can be properly handled by optimizing a tree-structured trajectory with each branch accounting for a potential hazard. While existing methods [9], [10], [11] have demonstrated the ability to handle stochastic interactions with other road users, their limited ability to handle a small number of surrounding agents may pose challenges for real-world applications. Moreover, existing approaches introduce strong assumptions on the upper-stream modules, making them poorly integrated into the existing autonomous driving stack.

To overcome these limitations, we proposed the MARC framework, a compact combination of behavioral and motion planning. We first generate critical scenario sets conditioned on semantic-level ego policies using forward simulations. By conducting the divergence assessment of future scenarios, we construct scenario trees with dynamic branchpoints. To diversify driving behaviors with tradeoffs between conservativeness and efficiency, we introduce *risk-aware contingency planning* (RCP) by advancing contingency planning with risk measurement. The proposed RCP generates trajectory trees considering diverse risk tolerances and future scenarios based on efficient bi-level optimizations with *linear programming* (LP) and the *iterative linear quadratic regulator* (iLQR). Finally, the policy selection is conducted using the evaluation of both scenario trees and trajectory trees, thereby minimizing the loss of multi-modal information. Illustrative examples are shown in Fig. 1. We demonstrate the promising ability of MARC through comparison with a strong baseline, quantitative ablation tests, and qualitative experiments on a self-built multi-agent platform and the open-source CARLA simulator [12]. The contributions of MARC include:

Manuscript received 28 April 2023; accepted 21 August 2023. Date of publication 30 August 2023; date of current version 6 September 2023. This letter was recommended for publication by Associate Editor H. Il Son and Editor H. Moon upon evaluation of the reviewers’ comments. This work was supported in part by the Hong Kong Ph.D. Fellowship Scheme, The Research Grants Council General Research Fund (RGC GRF) Project under Grant RMGS20EG20 and in part by the HKUST-DJI Joint Innovation Laboratory. (Corresponding author: Tong Li.)

Tong Li, Lu Zhang, and Shaojie Shen are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: tlibm@ust.hk; lzhangbz@ust.hk; eeshaojie@ust.hk).

Sikang Liu is with DJI Technology Company, Ltd., Shenzhen 518057, China (e-mail: sikang.liu@dji.com).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2023.3310431>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3310431

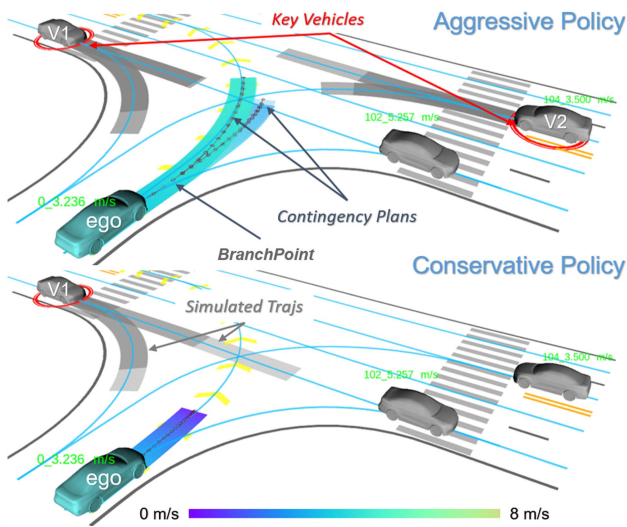


Fig. 1. Illustration of different policies generated by MARC. Policy-conditioned critical scenarios are rendered with different key vehicles. The darkness of trajectories indicates the associated probabilities. Scenario trees are adopted to depict the topology of rendered scenarios, and trajectory trees with risk-aware maneuvers are generated. The aggressive policy shows a siding action considering multi-modal interactions with the surrounding key vehicles, while the conservative policy converges to a smooth deceleration maneuver.

- MARC generates policy-conditioned critical scenarios and dynamically builds scenario trees that account for the scene-level divergence for each policy.
- Risk-aware contingency planning is introduced to efficiently generate diverse maneuvers which handle multi-modal interactions under different risk tolerances.
- We conducted extensive quantitative experiments, comparisons with strong baselines, and qualitative experiments to showcase the capability of MARC.

The remaining content is organized as follows. The related work is reviewed in Section II, and an overview of the proposed framework is presented in Section III. The methodologies are detailed in Sections IV and V. Implementation details and experimental results are provided in Sections VI and VII. Finally, the article is concluded in Section VIII.

II. RELATED WORK

A. Decision-Making Under Uncertainty

Extensive literature exists on addressing interactive planning under uncertainty in dynamic scenarios. POMDP provides a theoretically sound framework for handling multiple potential futures in a general form. However, due to the *curse of dimensionality*, solving POMDPs in real-time is infeasible as the driving context becomes more complex, even with advanced POMDP solvers [13], [14] and over-tailored problem modeling [2], [3], [4]. To further enable real-time decision-making, multipolicy-based methods are proposed to simplify the POMDP process using domain-specific knowledge of autonomous driving. Particularly, multipolicy-based methods prune the belief tree of the POMDP process using semantic-level policies which approximate the original action space and retain the interaction among agents using closed-loop forward simulations. MPDM [5], [6], a representative of multipolicy-based methods, shows a promising ability for interactive planning. Following the spirit of MPDM,

EUDM [7] considers action and intention branching during the planning horizon to enable more flexible decision-making in dynamic scenarios. The EPSILON framework extends forward simulation using advanced driver models to generate human-like behavior with improved safety, and is validated through real-world autonomous driving experiments [15]. TPP [16] utilizes deep learning prediction models to generate MDP with better scalability. Nevertheless, drawbacks exist. For instance, the multipolicy-based pipelines only consider policies that handle specific interactions in a single evolution and output the optimal policy over all scenarios, resulting in the lack of guarantee of policy consistency. Furthermore, existing methods only adopt single trajectory generation methods causing unavoidable loss of multi-modality information.

B. Motion Planning With Contingency

Motion planning is another extensively studied area in autonomous driving [17]. Techniques of motion planning can be categorized into search-based methods [18], [19], [20], optimization-based methods [21], [22], [23], and a combinations of these two [24], [25]. Being an optimization-based method, contingency planning generates deterministic actions for multiple future scenarios. In particular, it outputs trajectory trees with one shared segment and multiple branches to handle different problem setups independently. Using the model predictive control (MPC) to solve the branching trajectory tree with one shared state to track a desired path while considering the potential dangers, contingency MPC [26] shows the ability to counteract potential threats in static environments. Contingency planning can also generate deterministic actions, accounting for the motion uncertainty of other agents when combined with probabilistic obstacle predictions [9]. Branch MPC [10], [27] optimizes trajectory trees with multiple branchpoints based on scenario trees, taking into account the potential future branching of other agents' actions. Adapting similar ideas, reactive iLQR [11] generates flexible yet safe maneuvers accounting for future environments without rigorously avoiding the whole reachable regions of other agents. However, existing methods have limitations in generating effective interactions in complex circumstances with low computational costs, primarily due to the lack of guidance from the decision-making layer. Additionally, these methods' fixed or predefined scenario trees do not generalize well to real-world applications.

We address the above issues by systematically combining multipolicy decision-making and contingency planning. Our framework generates a scenario tree that extracts the critical driving context for each semantic-level policy. Based on the scenario tree, we achieve efficient risk-aware maneuver generations accounting for critical scenarios with user-defined risk tolerances leveraging contingency planning under risk measurement. The compact design of our method enables consistent policy generations under interaction uncertainty while minimizing information loss between layers.

III. SYSTEM OVERVIEW

As shown in Fig. 2, we present a complete planning pipeline that incorporates perception data from the simulator and generates optimal policy with trajectory trees for the vehicle controller. The vehicle controller evaluates certain target states on the trajectory trees and computes the corresponding control signals.

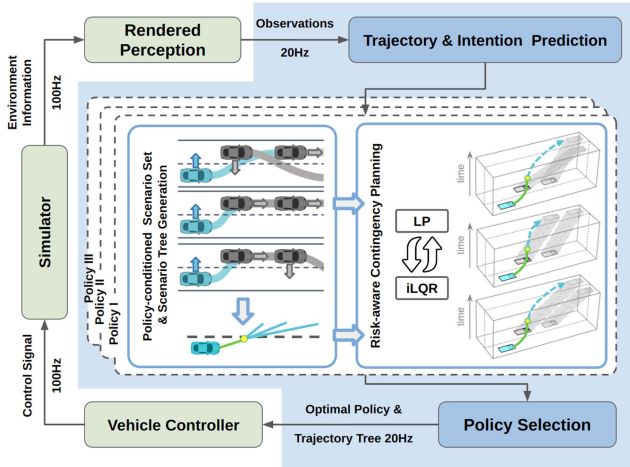


Fig. 2. Illustration of the MARC framework (marked in blue) and its relationship with other components. The ego vehicle is in cyan, and surrounding vehicles are in gray. In the scenario set, the cyan arrows indicate the ego policies while the gray arrows indicate the intentions of surrounding vehicles. The solid lines represent the simulated trajectories. In the scenario tree and trajectory tree, the branchpoint is in yellow, the shared part of the trajectories is in green, and the contingency plans are in cyan.

These control signals, along with those from other agents, are fed into the simulator, which synchronously propagates the states according to the underlying physical models.

MARC first predicts nearby vehicles' future trajectories and semantic-level intentions based on historical observations and environmental context. The critical scenario sets conditioned on the ego policies are then elected utilizing multi-agent forward simulation. The associated scenario trees with dynamic branch points are constructed utilizing scene-level divergence assessment. Taking the scenario trees, *risk-aware contingency planning* (RCP) generates optimal maneuvers with the user-defined risk tolerances. Specifically, RCP solves a bi-level optimization consisting of LP for risk measurement and iLQR for contingency planning. Leveraging the multi-modality in both behavioral and motion planning layers, MARC generates actions that consider both risk and multiple future evolutions.

IV. POLICY-CONDITIONED SCENARIO TREE

Using tree structures to represent scenarios provides a clear topology of future evolution. As elaborated in Section II, fixed or predefined scenario trees have poor generalization. In contrast, our approach involves constructing dynamic scenario trees that adapt to the change of interactions. The process is conducted in a two-step manner: generating policy-conditioned critical scenario sets via closed-loop forward simulations and building tree representations with scene-level divergence assessments.

We render scenarios based on the semantic-level ego policies and nearby agents' intention predictions. The computational complexity of the process is bounded by only considering policy-conditioned critical scenarios that might lead to potential risks. This pruning process is achieved by key vehicle selections with safety evaluations of non-cooperative behaviors for each intention combination. Following previous methods [6], [15], scenarios are rendered through closed-loop forward simulations using the selected vehicles with predicted intentions. Deterministic actions generated by the original forward simulation

assume all agents follow basic traffic rules and interact according to pre-defined models. However, hand-crafted models may diverge from reality, leading to unrealistic simulations in extreme circumstances. We adopt the idea of *forward reachable sets* (FRSs) [28], [29], which are used to provide safety guarantees for ego trajectory planning and extend them to capture the surrounding agents' potential tendencies in the simulation horizon. Therefore, the trajectory can generate open-loop responses swerving away from potential collision areas to ensure safety when closed-loop reactions are infeasible. To further ensure safety, we introduce fallback policies that take all FRSs as input and generate passive maneuvers such as pull-over and emergent braking. Hence, we ensure the completeness in the rendered scenario sets covering the agents' multi-modal behaviors conditioned on ego policies.

A typical scenario tree starts from the current state and branches into different evolution at the same timestamp, representing the *shared* and *independent* solution spaces of the rendered scenarios, respectively. Here, the process of building the scenario tree is primarily concerned with determining the *latest* timestamp at which the scenarios diverge. Delaying the branching time as much as possible offers two advantages. Firstly, the number of state variables in the trajectory optimization problem decreases as the duration of the shared part increase. More importantly, a later branching point gives the ego vehicle more reaction time to handle different potential outcomes smoothly. Since the ego trajectories are the cumulative results of the interaction with the scene context, their differences can be used to reveal the divergence of scenarios. Therefore, given the critical scenario set \mathcal{S} , the corresponding branch time calculation can be formally described as:

$$\begin{aligned} \max \tau \in \{0, \dots, \tau_{\max}\} \\ \text{s.t. } \forall \mathcal{M}(s_i, s_j, \tau) < \theta, \quad (s_i, s_j) \in \{\mathcal{S} \times \mathcal{S}\}, \end{aligned}$$

where \mathcal{M} is a function measuring the deviations between ego states at time τ given the scenario pairs (s_i, s_j) from the Cartesian product of the scenario set. The state divergence threshold is predefined as θ . To preserve the multi-modality and prevent the tree structure from collapsing into a single trajectory, a maximum branching time τ_{\max} is set to limit the length of the shared part. Despite its simple design, the proposed method is effective in practice. A brief depiction of the divergence assessment can be found in Fig. 3(a).

V. RISK-AWARE CONTINGENCY PLANNING

A. Prerequisites

Given the tree-structure descriptions of critical scenarios, we adopt contingency planning to cope with the multi-modality. As elaborated in Section II, contingency planning aims to obtain a trajectory tree that accounts for all possible surroundings evolutions. We define the planning problem with K contingency plans as a constrained nonlinear optimization:

$$\begin{aligned} \min_U \sum_{j \in I_s} l_j(x_j, u_j) + \sum_{k \in K} \sum_{j \in I_k} l_j(x_j, u_j) \\ \text{s.t. } x_i = f(\text{pre}(x_i), u_i), \quad \forall i \in I \setminus \{0\} \\ h_i(x_i, u_i) \leq 0, \quad \forall i \in I, \end{aligned}$$

where I , I_s , and I_k are the index lists of all nodes, shared nodes, and the nodes on the k -th contingency plan, respectively.

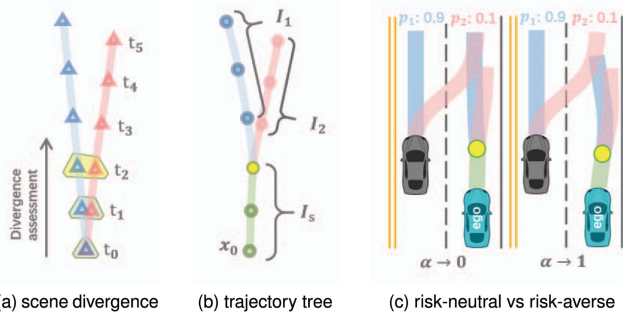


Fig. 3. (a) Depicts the branch time marked in yellow selected by divergence assessment. (b) Shows a trajectory tree, where I_s and I_k are the index lists of shared nodes, and the nodes on the k -th contingency plan, respectively. (c) Shows maneuvers generated by RCP with different risk tolerances α : risk-neutral (left) and risk-averse (right). The green trajectories are the shared trajectories. The branchpoints are colored in yellow. Contingency plans and associated predictions of agents are marked in the same colors.

$X := \{x_i | i \in I\}$ is the trajectory states while $U := \{u_i | i \in I \setminus \{0\}\}$ is the control signals. The cost function, constraint, and state-transition function are denoted by $l(\cdot)$, $h(\cdot)$, and $f(\cdot)$, respectively. $pre(\cdot)$ returns the input node's predecessor. Fig. 3(b) depicts a general trajectory tree.

B. Bi-Level Risk-Aware Contingency Planning

Due to the diverse risk tolerances, human drivers tend to react differently, even given the same surrounding predictions. Therefore, risk tolerances are essential for diverse maneuver generations. We adopt *conditional value-at-risk* (CVaR), which is recently introduced to contingency planning in [10], as a tail risk assessment measure in our framework. Given a discrete random variable of the distinct value set K with associated probabilities $\{p_k | k \in K\}$ and risk costs $\{\xi_k | k \in K\}$, the CVaR can be defined as follows [30]:

$$\begin{aligned} \text{CVaR}_\alpha &= \max_Q \sum q_k p_k \xi_k \\ \text{s.t. } 0 &\leq q_k \leq (1 - \alpha)^{-1}, k \in K \\ &\sum q_k p_k = 1, \end{aligned}$$

where $Q := \{q_k | k \in K\}$ is a series of variables that weigh the original risk cost under constraints. $\alpha \in [0, 1)$ is a hyperparameter that decides the percentage of worse cases that the risk measure should take into consideration.

To combine CVaR with contingency planning, we treat the safety part of cost functions l^{safe} as the risk costs, which will be weighted by the CVaR variables Q and leave the other cost functions l^{-safe} unbiased. The resulting *risk-aware contingency planning* (RCP) is shown as follow:

$$\begin{aligned} \max_Q \min_U &\sum_{j \in I_s} l_j(x_j, u_j) \\ &+ \sum_{k \in K} \sum_{j \in I_k} \left(p_k q_k l_j^{safe}(x_j, u_j) + l_j^{-safe}(x_j, u_j) \right) \\ \text{s.t. } &x_i = f(pre(x_i), u_i), \quad \forall i \in I \setminus \{0\} \\ &h_i(x_i, u_i) \leq 0, \quad \forall i \in I \\ &0 \leq q_k \leq \alpha^{-1}, k \in K \end{aligned}$$

$$\sum q_k p_k = 1,$$

By changing α , which indicates the risk tolerance levels of users, we can obtain diverse driving behaviors through RCP. As shown in Fig. 3(c), if setting $\alpha \rightarrow 0$, we can obtain risk-neutral maneuvers with the optimal variables Q closed to given probabilities. Conversely, choosing $\alpha \rightarrow 1$ brings risk-averse maneuvers with increased optimal variables Q weighting on the dangerous contingency plans.

The above problem can be solved in its dual form by introducing dual variables [10]. However, this method increases the problem dimension and scales poorly to the high-dimensional state-transition function. To further improve the computational efficiency, we adopt the bi-level optimization method in which the solution to the original problem is obtained by iteratively solving two subproblems using iLQR and LP.

iLQR utilizes the Taylor-series expansion to obtain approximated problems consisting of quadratic cost functions and discrete system transition functions of the original problem. Given fixed risk-related variables Q , we can obtain the approximated subproblem for control sequence U as follows:

$$\begin{aligned} \min_U &\sum_{j \in I_s} \left(\frac{1}{2} x_j^T R_j x_j + r_j^T x_j + \frac{1}{2} u_j^T S_j u_j + s_j^T u_j \right) \\ &+ \sum_{k \in K} \sum_{j \in I_k} p_k q_k \left(\frac{1}{2} x_j^T R_j^{safe} x_j + r_j^{safe T} x_j \right) \\ &+ \sum_{k \in K} \sum_{j \in I_k} \left(\frac{1}{2} x_j^T R_j^{-safe} x_j + r_j^{-safe T} x_j + \frac{1}{2} u_j^T S_j u_j + s_j^T u_j \right) \\ \text{s.t. } &x_i = A_i pre(x_i) + B_i u_i, \quad \forall i \in I \setminus \{0\}, \end{aligned}$$

where $A_i \equiv \frac{\partial f}{\partial pre(x_i)}$ and $B_i \equiv \frac{\partial f}{\partial u_i}$. This approximated problem could be solved by iLQR in the *discrete dynamic programming* style using backward and forward propagation. We refer the interested readers to [31].

Note that for any fixed control sequence U , the original problem degenerates into an LP. Solving the LP finds the CVaR with the optimal variables Q under an extra equality constraint. For any fixed Q , the original problem can be approximated by iLQR. Solving the iLQR subproblem obtains optimal controls U of the trajectory tree. Since both subproblems are convex, we can derive a bi-level optimization with a convergence guarantee:

$$\begin{aligned} U^{k+1} &= \underset{U}{\operatorname{argmin}} \text{iLQR}(X^k, U^k, Q^k) \\ Q^{k+1} &= \underset{Q}{\operatorname{argmax}} \text{LP}(X^{k+1}, U^{k+1}, Q^k). \end{aligned}$$

Owing to the low cost of solving LP and iLQR, the computational efficiency of the risk-aware contingency planning can be guaranteed. Compared with original contingency planning, RCP generates trajectories that are optimal in multiple future scenarios under user-defined risk-averse levels, which can give rise to more human-like behaviors.

C. Policy Evaluation

As stated in Section II, existing multipolicy-based pipelines usually require a subsequent motion planner that only considers a single selected scenario, leading to potential oscillations in the final output when dealing with rapidly changing scenarios. In contrast, our framework conducts policy selection based on the

Algorithm 1: Process of MARC.

Inputs : Perception Z , Ego vehicle state x , Ego policy set Π , Horizon τ_h , Max Branch Time τ_{max} , Divergence threshold θ

Outputs: Optimal policy π^* , Optimal trajectories \mathcal{T}^*

```

1  $\mathcal{R} \leftarrow \emptyset$  ; //init policy reward set
2  $\mathcal{T} \leftarrow \emptyset$  ; //init policy trajectory set
3  $\mathcal{P}, \mathcal{I} \leftarrow \text{PredictIntentionTrajectories}(Z, x)$  ;
4  $\mathcal{I}_c \leftarrow \text{GetIntentionCombination}(\mathcal{I})$  ;
5 for  $\pi \in \Pi$  do
    /* Scenario Generation */
6    $\mathcal{S}_\pi \leftarrow \emptyset$  ; //init scenario set
7   for  $c \in C$  do
8      $V_k \leftarrow \text{GetKeyVehicles}(Z, c, x, \pi)$  ;
9      $\mathcal{S}_\pi \leftarrow \text{ForwardSimulation}(Z, c, V_k, x, \pi, \tau_h)$  ;
10  end
11   $\Psi_\pi \leftarrow \text{ScenarioTreeConstruction}(\mathcal{S}_\pi, \tau_{max}, \theta)$  ;
12   $\mathcal{T}_\pi \leftarrow \text{RcpOptimization}(\pi, \Psi_\pi, \tau_h)$  ;
13   $\mathcal{T} \leftarrow \mathcal{T}_\pi$  ;
14   $\mathcal{R} \leftarrow \text{PolicyEvaluation}(\pi, \Psi_\pi, \mathcal{T}_\pi)$  ;
15 end
16  $\pi^*, \mathcal{T}^* \leftarrow \text{PolicySelection}(\mathcal{R}, \mathcal{T})$  ;

```

evaluation of both scenario trees and trajectory trees, ensuring the consistency of the resulting policies.

The flow of the MARC framework is detailed in Algorithm 1. Policy-conditioned scenarios are rendered under each policy based on the intention predictions (Line 7 to Line 10). The scenario trees and trajectory trees are obtained consecutively from the critical scenario sets (Line 11 to Line 12). Evaluations are conducted based on both scenario trees and trajectory trees (Line 14). Most parts of our framework (Line 5 to Line 15) can be run in parallel for real-time performance.

VI. IMPLEMENTATION DETAILS

A. Trajectory and Intention Prediction

We adopt a neural network-based motion predictor based on [32] to provide scene-centric multi-modal trajectory prediction for all nearby agents. The network is trained on the Argoverse dataset [33] and deployed in C++ via LibTorch.¹ After obtaining the predicted trajectories and their probability scores, the next step is to transform them into semantic-level intentions. In this work, the longitudinal intentions of other agents are defined as a set of semantic actions (i.e., maintaining speed, acceleration, and deceleration), while the lateral intentions are represented as the reference lane sequences. To extract the lateral intentions, a simple voting method is used based on the predicted trajectories' positional distribution. In particular, we begin by extracting candidate reference lanes for the target agent and then associate each predicted trajectory with these candidate lanes to evaluate the likelihood of each lane. Once the lateral intention is obtained, we use a similar method to estimate the longitudinal intentions and associated probabilities. We note that it is possible to predict both future trajectories and intentions using a single network, which we leave as future work.

¹The C++ distribution of PyTorch. See <https://pytorch.org> for details.

B. Ego Policy Design

Considering achieving diverse driving behaviors, we define semantic-level policies for the ego vehicle, which can be categorized as longitudinal and lateral policies. Both longitudinal and lateral policies encompass basic semantic-level policies and scene-related policies. For example, in urban driving, the basic longitudinal policies could be maintaining speed, decelerating, and accelerating. Scene-related policies could be stopping at a specific position (i.e., stop line and conflict zone), decelerating to yield, and accelerating to the speed limit. Similarly, the lateral policies can be defined as basic policies such as lane keeping, lane change to the target lane sequence, and scene-related policies such as bypass and in-lane siding.

C. Forward Simulation With FRSSs

Existing work usually obtains reachable sets with model-based propagation or over-approximations [34], [35]. In this work, we adopt a simple yet effective method to generate FRSSs. Specifically, we directly compute the spatial-temporal occupancy of multi-modal predictions and collect the occupancy union according to the timestamp to form the FRSSs. Since FRSS is essentially a spatial-temporal occupation description without probabilistic features and the trajectory predictions have implicitly encoded the estimation of the hidden state as well as the transition model of the surrounding agents, our method can approximate these occupations with little computational cost while preserving multi-modality.

The forward simulation takes two steps. We first conduct the deterministic closed-loop forward simulation similar to [7] with safety assessments on the generated trajectories. The failure of the assessments will trigger the second simulation where we replace those failed policies with our fallback policies. By running the second forward simulation, we can obtain the final trajectories with closed-loop reactions to nearby agents and open-loop reactions to FRSSs.

D. iLQR Design

We adopt the kinematic bicycle model as the state-transition function $f(\cdot)$ in iLQR and design the cost function of i -th node l_i as the sum of safety cost l_i^{safe} , target cost l_i^{tar} , kinematic cost l_i^{kin} and comfort cost l_i^{comf} :

$$l_i = l_i^{safe} + l_i^{tar} + l_i^{kin} + l_i^{comf}.$$

Similar to [23], environmental constraints can be modeled utilizing vectorized representations (such as polygons for drivable areas and polylines for reference lines). We introduce a distance function $\mathcal{D}(\cdot)$ that returns the Euclidean distance between the ego vehicle and the environmental constraint. To ensure the second-order continuity, we further define a smooth function with parameter β as follows:

$$\mathcal{S}_\beta(\Gamma) = \sum_{i=1}^{\Omega} \frac{\exp(\beta\gamma_i)\gamma_i}{\sum_{i=1}^{\Omega} \exp(\beta\gamma_i)}, \Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_\Omega\}^T.$$

With a slight abuse of notation, we denote $\mathcal{G}(\cdot)$ as the user-designed penalty function. The safety cost penalizes the signed distances to multiple environmental constraints:

$$l_i^{safe} = \overbrace{\mathcal{G}_{driv}((1 - \mathbf{1}_{P_d}) \mathcal{S}_{\beta < 0}(\mathcal{D}(P_d)))}^{\text{drivable area}}$$

$$\begin{aligned}
 & + \sum_{j \in n_p} \overbrace{\mathcal{G}_{bb} \left(\mathbf{1}_{P_{b_j}} \mathcal{S}_{\beta < 0}(\mathcal{D}(P_{b_j})) \right)}^{\text{bounding box}} \\
 & + \sum_{j \in n_p} \overbrace{\mathcal{G}_{rs} \left(\mathbf{1}_{P_{r_j}} \mathcal{S}_{\beta < 0}(\mathcal{D}(P_{r_j})) \right)}^{\text{reachable set}},
 \end{aligned}$$

where P_d defines a drivable area that is feasible and collision-free to static obstacles, P_{b_j} is the i -th inflated bounding boxes of j -th surrounding agent, P_{r_j} is the FRS of j -th surrounding agent and $\mathbf{1}(\cdot)$ denotes the enclosed indicator function. Similar formulations can be found in the domain of backup policy control barrier function quadratic programming [36].

The target term varies depending on the policies and the scenarios. For instance, the reference can be a centerline that serves as a guide for lane change policies in highway scenarios, while alternatively representing a desired area for overtaking at an intersection. For simplicity, we divide this term into two parts: reference cost and desired area cost:

$$l_i^{tar} = \mathcal{G}_{ref}(\mathcal{S}_{\beta < 0}(\mathcal{D}(L_{ref}))) + \mathcal{G}_{des}(\mathcal{S}_{\beta < 0}(\mathcal{D}(P_{des}))),$$

where L_{ref} is the reference polyline of target lane sequence, P_{des} is the target area defined by polygon.

The kinematic constraints are introduced by penalizing the state and control variables that exceed the limits:

$$\begin{aligned}
 l_i^{kin} = & \mathcal{G}_{xub}(max(x_t - x_{ub}, 0)) + \mathcal{G}_{xlb}(max(x_{lb} - x_t, 0)) \\
 & + \mathcal{G}_{uub}(max(u_t - u_{ub}, 0)) + \mathcal{G}_{ulb}(max(u_{lb} - u_t, 0)),
 \end{aligned}$$

where x_{ub} , x_{lb} are the upper and lower bound of the state variables, while u_{ub} , u_{lb} are the upper and lower bound of the control inputs, $max(\cdot)$ is the element-wise max function.

The final term of the objective function is designed to minimize both the longitudinal discomfort (i.e., acceleration and jerk) and the lateral discomfort (i.e., lateral acceleration and steering rate), thereby ensuring the smoothness of the generated trajectories:

$$l_i^{comf} = \mathcal{G}_{lon}(x_t) + \mathcal{G}_{lat}(x_t).$$

E. Policy Evaluation

In this work, the total policy reward is evaluated based on the scenario contexts and risk-aware contingency plans. Specifically, the reward of policy is defined as the negative weighted sum of several components:

$$R = -(\lambda_1 F_s + \lambda_2 F_e + \lambda_3 F_n + \lambda_4 F_r + \lambda_5 F_u),$$

where the safety cost F_s is measured by the distance between trajectories of ego and other vehicles. The efficiency cost F_e is computed from the gap between the average velocity of contingency plans and desired velocity. The navigation cost F_n is obtained by the difference between the routing and target lane sequence. The risk cost F_r is obtained from the cost of RCP, and the uncertainty cost F_u is measured by the branch time and the divergence between different contingency plans.

VII. EXPERIMENTAL RESULTS

A. Simulation Platform and Environment

The experiments are conducted on a self-built multi-agent simulation platform elaborated in Section III and CARLA. The

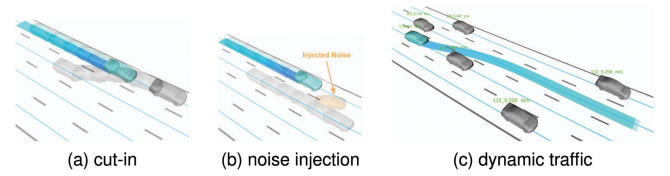


Fig. 4. Illustration of the scenario setups. (a) Shows the snapshots of the cut-in scenario. The ego vehicle is expected to yield to the non-cooperative vehicle. (b) Illustrates the noise injection (marked in yellow) on the predictions, (c) is a typical dynamic traffic scenario in which the ego vehicle has to interact with other agents under uncertainty.

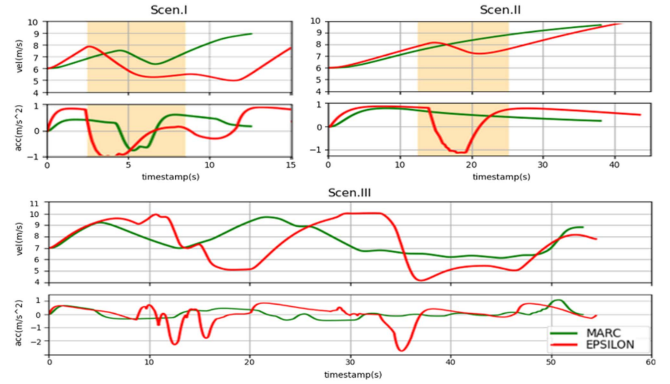


Fig. 5. Dynamic profiles of the comparisons with EPSILON. In Scen.I, MARC captures the cut-in intention and generates smoother deceleration during the cut-in period marked in yellow. In Scen.II, the additional noise is added to the predictions from 2.5 s to 6 s marked in yellow. MARC handles the disturbances while EPSILON switches the policy which results in uncomfortable deceleration. In Scen.III, MARC outperforms EPSILON with better efficiency and riding comfort.

proposed MARC framework is implemented in C++11 with customized solvers. All experiments are run on a desktop computer equipped with Intel i9-12900 K CPU and 32 GB RAM with a stable running rate of 20 Hz.

B. Quantitative Results

We conduct comparisons with the existing strong baseline and ablation tests on our self-built platform to evaluate the performance of the proposed method quantitatively.

1) *Comparison With EPSILON*: We conduct qualitative comparisons with the existing strong baseline EPSILON in the three scenarios. Scen.I is a non-cooperative cut-in scenario aiming to test the ability to handle extreme cases. Scen.II is a noise injection test where the adjacent vehicle remains lane-keeping during the test while noise is injected into the prediction module of the ego vehicle, increasing the lane-changing prediction, as shown in Fig. 4(b). This setup simulates the case where predictions deviate from the ground truth, which is common in real-world applications. Scen.III is a highway scenario with multiple autonomous agents, which is set up to assess the planning ability under intensive uncertain interactions. As shown in Fig. 4(c), the ego vehicle is assigned a global route and a speed limit to navigate through the traffic in which agents will conduct various behaviors. For fair comparisons, we use the same prediction module for both methods. To show that the policies of MARC are more robust under uncertain interactions and bring fewer unexpected policy switches, which results in better efficiency and comfort, we use total time and average

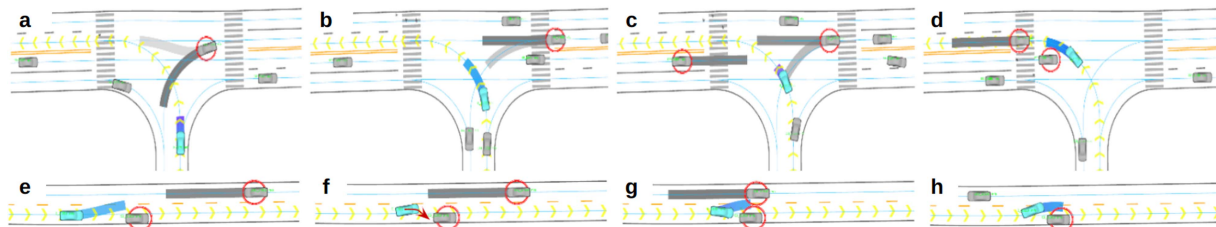


Fig. 6. Key frames of the qualitative experiments. The global route is marked by yellow arrows, key vehicles are marked by red circles, and the simulated trajectories are marked by gray bands with darkness representing probabilities. (a)–(d) Unprotected left turn: ego vehicle first yields to the aggressive left-turning vehicle (a), slowly moves forward to stop the oncoming vehicle on the left meanwhile yields to the second vehicle on the right (b) and (c), and finally pass around the stopped vehicle to exit the intersection (d). (e)–(h) Bypass with oncoming traffic: ego vehicle first swerves to the left, trying to bypass the parked vehicle (e), once anticipating the oncoming vehicle is not likely to yield, ego vehicle aborts bypassing and turn into the original lane for collision avoidance (f), and it quickly recovers the bypass policy as soon as the spatiotemporal constraints are met (g) and (h).

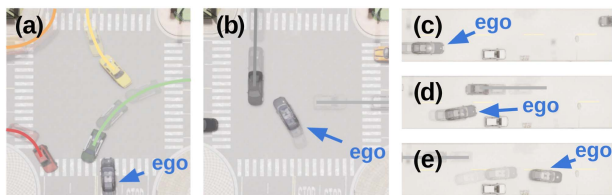


Fig. 7. Keyframes of tests on CARLA. The historical positions and trajectories of surrounding agents are marked with transparent colors. (a) and (b) Ego vehicle achieves human-like defensive left-turning behaviors in intersection with random traffic. (c)–(e) Ego vehicle achieves bypassing with negotiation behaviors.

TABLE I
COMPARISON WITH EPSILON

Methods		Time (s)	Avg Spd (m/s)	RMS Acc (m/s ²)	Max Abs Acc (m/s ²)
Scen. I	EPSILON	19.90	6.87	0.63	1.03
	MARC	12.56	7.26	0.45	0.75
Scen. II	EPSILON	8.70	7.90	0.72	1.13
	MARC	7.60	8.03	0.51	0.78
Scen. III	EPSILON	54.46	7.41	0.70	2.78
	MARC	53.13	7.61	0.34	1.09

speed during the test for efficiency measurement and use the root mean squared acceleration and maximum absolute acceleration for comfort assessment [37], [38]. As shown in Table I, our method outperforms the baseline with improvements in all metrics. The dynamic profiles are shown in Fig. 5 to show side-by-side comparisons in three scenarios. MARC, being more cautious when facing multiple predictions, accelerates smoother and decelerates in time in Scen.I. In Scen.II, MARC executes more consistent policies under noisy predictions, whereas EPSILON switches its policies resulting in sudden decelerations. In Scen.III, MARC handles the uncertain interactions better than EPSILON, with fewer unnecessary decelerations and a higher average speed, improving comfort and efficiency.

2) *Ablative Tests of RCP*: We modify the cut-in scenario Fig. 4(a) for the ablation test, in which the ego vehicle only executes the lane-keeping policy, and the neighboring vehicle conducts a non-cooperative lane-change maneuver forcing the ego vehicle to yield. A baseline planner, a fixed branching mechanism, a dynamic branching mechanism, and a dynamic branching mechanism with risk-aware optimization methods will be tested in this scenario 100 times with random initialization. We use scene-specific metrics for comfort and safety assessments: average maximum deceleration, average

TABLE II
ABLATION STUDY RESULTS

Methods	Avg Max Dec (m/s ²)	Avg Min Dis (m)	Suc Rate (%)
w/o branch	3.48	1.37	83
Fixed branch	2.52	2.53	93
Dyna branch	1.63	3.45	96
Dyna branch + Risk	1.14	4.12	100

minimum distance with respect to the cut-in vehicle, and success rate (non-collision cases vs. total cases). Metrics such as time to collision and intervehicular time are also applicable for safety assessments [39]. As shown in Table II, the baseline only reacts after the most likely prediction aligns with the cut-in behaviors, resulting in the hardest deceleration, closest distance, and the lowest success rate. Branching mechanisms enables planners to respond earlier, smoothing the deceleration and increasing the minimum distance. Dynamic branching outperforms fixed one owing to the better adaptation to the scenarios. Combining risk measures with dynamic branching brings a more defensive yet smoother driving style which achieves the best performance among methods, proving the effectiveness of RCP.

C. Qualitative Results

We conduct qualitative tests in highly interactive scenarios such as unprotected left turns and bypass with oncoming traffic on the self-built platform and scenarios with random traffic on CARLA to further verify the generalization of our method.

1) *Unprotected Left Turn*: The ego vehicle enters a 3-way intersection with no traffic signals. Vehicles from different directions are randomly generated, and their actual intentions are unknown to the ego vehicle. As shown in Fig. 6(a)–(d), the ego vehicle first slows down to yield to the first left-turning vehicle (a), creeps forward to stop the vehicle on the left while giving the right of way to the second vehicle on the right (b) and (c), and finally exits the intersection with a swerving maneuver to avoid the potential collision (d). The result shows MARC’s ability to handle uncertain intentions and generate reasonable behaviors during negotiations.

2) *Bypass With Oncoming Traffic*: This scenario consists of a parked vehicle and a fast-moving vehicle in the opposite lane. As shown in Fig. 6(e)–(h), the ego vehicle first swerves to the left, attempting to conduct a left bypassing maneuver on the parked vehicle (e), once noticing that the oncoming agent is less likely to yield, it aborts bypassing policy and returns the original

lane for collision avoidance (f), and after holding to yield for a while, it switches back to the bypass policy once it anticipates the spatial-temporal constraints are satisfied (g) and (h). The result reveals MARC's ability to efficient behavior generations under spatial-temporal constraints in conflict situations.

3) *Qualitative Tests on CARLA*: We extend our method to CARLA and conduct tests on random traffic flows. As shown in Fig. 7, MARC generates human-like driving behaviors such as defensive left-turning and bypassing under negotiation in the tests, proving its generalization in various scenarios.

VIII. CONCLUSION AND FUTURE WORK

We present the MARC framework as a solution for autonomous driving in dynamic interactive environments. Our framework systematically unified two innovative techniques, policy-conditioned scenario tree and risk-aware contingency planning, to generate safe and interactive driving behaviors considering risk tolerance and uncertainty. Comparisons with a strong baseline and comprehensive experiments in challenging scenarios proved the effectiveness of our design. In the future, we will conduct real-world field tests for MARC.

REFERENCES

- [1] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1/2, pp. 99–134, 1998.
- [2] 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 *Proc. IEEE Intell. Veh. Symp.*, 2015, pp. 1126–1133.
- [3] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 1, pp. 5–17, Mar. 2018.
- [4] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "PORCA: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3418–3425, Oct. 2018.
- [5] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1670–1677.
- [6] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," *Auton. Robot.*, vol. 41, no. 6, pp. 1367–1382, 2017.
- [7] L. Zhang, W. Ding, J. Chen, and S. Shen, "Efficient uncertainty-aware decision-making for automated driving using guided branching," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 3291–3297.
- [8] P. O. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Control*, vol. 43, no. 8, pp. 1136–1142, Aug. 1998.
- [9] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 913–929, Aug. 2013.
- [10] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin, and A. D. Ames, "Interactive multi-modal motion planning with branch model predictive control," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5365–5372, Apr. 2022.
- [11] F. Da, "Comprehensive reactive safety: No need for a trajectory if you have a strategy," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 2903–2910.
- [12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.
- [13] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," *J. Artif. Intell. Res.*, vol. 58, pp. 231–266, 2017.
- [14] H. Kurniawati and V. Yadav, "An online POMDP solver for uncertainty planning in dynamic environment," in *Proc. Int. Symp. Robot. Res.*, Springer, 2016, pp. 611–629.
- [15] W. Ding, L. Zhang, J. Chen, and S. Shen, "EPSILON: An efficient planning system for automated vehicles in highly interactive environments," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1118–1138, Apr. 2022.
- [16] Y. Chen, P. Karkus, B. Ivanovic, X. Weng, and M. Pavone, "Tree-structured policy planning with learned behavior models," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 7902–7908.
- [17] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
- [18] M. Rufli and R. Siegwart, "On the design of deformable input-/state-lattice graphs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3071–3077.
- [19] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4889–4895.
- [20] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1961–1976, Aug. 2015.
- [21] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 2061–2067.
- [22] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha—A local, continuous method," in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 450–457.
- [23] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative LQR for on-road autonomous driving motion planning," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–7.
- [24] T. Gu, J. Atwood, C. Dong, J. M. Dolan, and J.-W. Lee, "Tunable and stable real-time trajectory planning for urban autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 250–256.
- [25] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 4327–4332.
- [26] J. P. Alsterda, M. Brown, and J. C. Gerdes, "Contingency model predictive control for automated vehicles," in *Proc. Amer. Control Conf.*, 2019, pp. 717–722.
- [27] R. Wang, M. Schuurmans, and P. Patrinos, "Interaction-aware model predictive control for autonomous driving," in *Proc. Eur. Control Conf.*, 2023, pp. 1–6.
- [28] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots," *Int. J. Robot. Res.*, vol. 39, pp. 1419–1469, Aug. 2020.
- [29] S. Vaskov, U. Sharma, S. Kousik, M. Johnson-Roberson, and R. Vasudevan, "Guaranteed safe reachability-based trajectory design for a high-fidelity model of an autonomous passenger vehicle," in *Proc. Amer. Control Conf.*, 2019, pp. 705–710.
- [30] R. T. Rockafellar, S. P. Uryasev, and M. Zabarankin, "Deviation measures in risk analysis and optimization," *Univ. Florida, Dept. Ind. & Syst. Eng.*, Paper 2002-7, Dec. 22, 2002. [Online]. Available: <https://ssrn.com/abstract=365640>
- [31] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *Int. J. Control*, vol. 3, no. 1, pp. 85–95, 1966.
- [32] T. Zhao et al., "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12126–12134.
- [33] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8748–8757.
- [34] S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Vasudevan, "Safe trajectory synthesis for autonomous driving in unforeseen environments," in *Proc. Dyn. Syst. Control Conf.*, vol. 58271, 2017, Art. no. V001T44A005.
- [35] I. Hwang, D. M. Stipanovic, and C. J. Tomlin, "Applications of polytopic approximations of reachable sets to linear dynamic games and a class of nonlinear systems," in *Proc. Amer. Control Conf.*, 2003, pp. 4613–4619.
- [36] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 6835–6841.
- [37] C. Pek and M. Althoff, "Fail-safe motion planning for online verification of autonomous vehicles using convex optimization," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 798–814, Jun., 2021.
- [38] M. Elbanhawi, M. Simic, and R. Jazar, "In the passenger seat: Investigating ride comfort measures in autonomous cars," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 3, pp. 4–17, Fall 2015.
- [39] S. Glaser, B. Vanholme, S. Mammari, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 589–606, Sep. 2010.