TUDelft

Delft University of Technology

An Efficient Game-Theoretic Planner for Automated Lane Merging with Multi-Modal Behavior Understanding

Zhang, L.; Han, S.; Grammatico, S.

Citation (APA)
Zhang, L., Han, S., & Grammatico, S. (2023). An Efficient Game-Theoretic Planner for Automated Lane Merging with Multi-Modal Behavior Understanding. In *Proceedings of the IEEE 26th International Conference on Intelligent Transportation Systems, ITSC 2023* (pp. 3085-3090). (IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC). IEEE. https://doi.org/10.1109/ITSC57777.2023.10422316

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

An Efficient Game-Theoretic Planner for Automated Lane Merging with Multi-Modal Behavior Understanding

Luyao Zhang[1,†], Shaohang Han[2,†] and Sergio Grammatico[1]

Abstract— In this paper, we propose a novel behavior planner that combines game theory with search-based planning for automated lane merging. Specifically, inspired by human drivers, we model the interaction between vehicles as a gap selection process. To overcome the challenge of multi-modal behavior exhibited by the surrounding vehicles, we formulate the trajectory selection as a matrix game and compute an equilibrium. Next, we validate our proposed planner in the high-fidelity simulator CARLA and demonstrate its effectiveness in handling interactions in dense traffic scenarios.

I. INTRODUCTION

Automated vehicles are facing a significant challenge when navigating in highly interactive environments, such as the lane-merging scenario shown in Figure 1. Traditional methods typically adopt a hierarchical structure where motion prediction and planning are decoupled. Consequently, these methods might be overly conservative since they often overlook the mutual interaction between the ego vehicle and the surrounding ones. Although newly developed learning-based approaches [1], [2] consider such interaction, they typically require large amounts of data and might lack interpretability. Another popular interaction-aware method is the Partially Observable Markov Decision Process (POMDP) [3], which provides a rigorous mathematical framework for handling incomplete information, such as the unknown intentions of the surrounding vehicles. However, solving a large-scale POMDP is computationally intractable.

As an approximation of the POMDP framework, the multiple policy decision-making (MPDM) [4] and its extension EPSILON [5] have demonstrated promising results in generating practically reasonable trajectories while remaining computationally efficient. The approach involves conducting multi-vehicle forward simulations based on semantic-level policies, followed by a trajectory evaluation step to select the best trajectory using handcrafted criteria. However, the rule-based trajectory evaluation in EPSILON might be overly aggressive or conservative when the surrounding vehicles have multiple behavior modes. Additionally, although the open-loop planning strategy is computationally efficient, it sacrifices the advantages of active information gathering in the original POMDP approach.

To systematically evaluate the trajectories, we use game theory, which is a powerful mathematical framework that
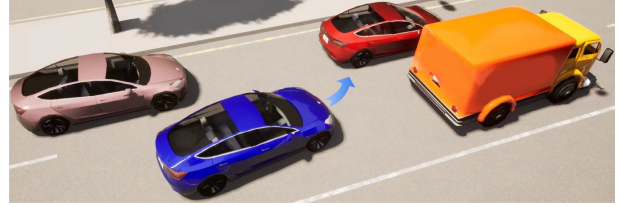
Fig. 1: Lane-merging scenario in the CARLA simulator. The ego vehicle (blue) is merging onto a lane in dense traffic.

captures the mutual influence between multiple agents. Previous research has explored equilibrium solutions extensively for automated lane merging. Some studies have focused on jointly planning trajectories for all vehicles by seeking a Nash equilibrium [6], [7]. However, these methods can only find a local equilibrium, and the quality of the solution might heavily depend on the initial guess. In contrast, other approaches use semantic-level actions as strategies. Among them, some studies propose a Stackelberg game with a leader-follower structure [8], [9]. However, determining the relative role of the leader or follower might be difficult [8]. In contrast to the leader-follower structure in a Stackelberg game, a Nash game treats all agents equally. A representative method based on a Nash game is proposed in [10], but it lacks validation in a high-fidelity simulator.

Contribution: In this paper, we propose a novel approach that combines game theory with interactive trajectory generation. To make the algorithm practical and efficient, we leverage the semantic-level actions and model the interaction between vehicles as a gap selection process (Section IV-A). Additionally, to tackle the issue of multi-modality, we represent the behavior of the surrounding vehicles as actions in a matrix game, and then select the Nash equilibrium with the lowest social cost [11] (Section IV-C). We also investigate the existence of Nash equilibria and the relationship between Nash and Stackelberg equilibria through both theoretical analysis and numerous numerical simulations (Section V and VI-A). Moreover, we validate the effectiveness of the proposed planner in the high-fidelity CARLA simulator [12] (Section VI-B).

II. RELATED WORK

POMDP for lane merging. Prior research has explored POMDP for addressing the issue of unknown intentions in lane-merging scenarios. Online solvers, such as POMCPOW [13] and DESOPT [14], estimate the action-value function through sampling. Other POMDP approximations, such as QMDP [15] and heuristics [16], have been proposed as well.
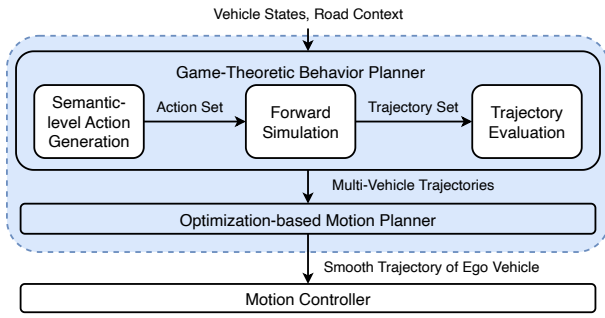
Fig. 2: Structure of the proposed behavior and motion planner. The focus of this paper is on the game-theoretic behavior planner.

Game-theoretic planning for lane merging. In methods that use Stackelberg games, vehicle strategies are represented by semantic-level actions such as motion primitives [8] or waiting time before merging [9]. While most of these methods assume the ego vehicle to be the leader, the rationale behind this is not always clear. Another approach utilizes level-k reasoning to model human drivers' behavior [17]. However, the computational burden of this framework is substantial due to the necessity of modeling the depth of human thinking [18].

III. PROBLEM FORMULATION

In this paper, we consider a mixed-traffic scenario where an automated ego vehicle interacts with the surrounding vehicles, as shown in Figure 1. Specifically, the ego vehicle aims at driving efficiently but a low-speed vehicle (SV0) travels in front of it. To avoid being blocked, the decision-making system of the ego vehicle needs to consider the diverse driving behaviors of the surrounding vehicles, select a suitable merging gap, and determine if/when to change lane. For example, in Figure 1, the ego vehicle can merge ahead of or after the pink vehicle (SV2). In fact, if SV2 yields, then the gap enlarges and the ego vehicle merges ahead of SV2. Otherwise, if the gap is not sufficiently wide, then the ego vehicle might slow down and then merge after SV2.

A. Structure of the Planner

We propose a game-theoretic planner as illustrated in Figure 2. Unlike other conventional behavior planners that require a motion predictor as an upstream module, in our approach, we combine motion prediction and behavior planning. Our proposed game-theoretic behavior planner consists of three modules: semantic-level action generation, forward simulation, and trajectory evaluation. First, we enumerate the possible semantic-level decision sequences of the traffic participants over the decision horizon. For the ego vehicle, a semantic-level decision can be making a lane change, accelerating or decelerating. Then, we form the action tuples by combining the decision sequences of the ego vehicle and surrounding vehicles. For each action tuple, the forward simulator generates the trajectories of the relevant vehicles. Subsequently, the trajectory evaluator determines the costs of the trajectories for each action tuple. Next, we construct a matrix game and seek an equilibrium. Finally, the trajectory

evaluator selects the action tuple associated with the equilibrium, and outputs the multi-vehicle trajectories.

Although we apply the kinematic bicycle model to simulate the motion of the ego vehicle, the trajectory generated by the behavior planner might not be sufficiently smooth due to the coarse discretization step. Therefore, we employ an additional local motion planner [19] to produce a kinematically feasible trajectory. To ensure safety, the simulated trajectories of the surrounding vehicles are used to impose dynamic collision avoidance constraints.

B. Matrix Game

We model the decision-making process as a matrix game with two players. In the game, each player selects an action from its finite action set to optimize its individual cost. A two-player matrix game is defined by a tuple $(\mathcal{N}, \Pi, J)$, where $\mathcal{N}$ is the set of two players, $\Pi = \times_{i \in \mathcal{N}} \Pi_i$ is the joint action space, and $J = \times_{i \in \mathcal{N}} J_i$ is the joint cost function. Next, we introduce the three ingredients of the matrix game: players, actions and cost functions.

1) Players: We consider the ego vehicle (EV) and the group of the surrounding vehicles (SV) as two players, $\mathcal{N} := \{EV, SV\}$.

2) Actions: Human drivers typically make semantic-level decisions to make lane changes safely and efficiently. Inspired by human drivers, we represent the action of player $i \in \mathcal{N}$ by a semantic-level decision sequence, denoted as $\pi_i = \{\pi_{i,0}, \ldots, \pi_{i,k}, \ldots, \pi_{i,H-1}\}$, where H is the decision horizon. We provide more design details on the decision sets and the method for enumerating all possible decision sequences later in Section IV-A.

3) Cost functions: Before computing the costs, the forward simulator (Section IV-B) generates multi-vehicle trajectories. The cost function J_i of vehicle i evaluates the corresponding trajectory based on user-defined metrics, such as safety, efficiency, comfort and navigation. We consider the surrounding vehicles as a whole by calculating the total cost as $J_{SV} := \sum_{i=2}^{N} J_i$, where N is the number of vehicles. Technical details are provided in Section IV-C.

A matrix game is depicted in Table I, where each entry represents a cost tuple $(J_{SV}^{ij}, J_{EV}^{ij})$ received by the group (SV) and the ego vehicle (EV) after performing their respective actions, π_{SV}^i and π_{EV}^j. Next, we look for an equilibrium

TABLE I: Game in normal form.

	π_{EV}^1	$\cdots$	$\pi_{EV}^{M_{EV}}$
π_{SV}^1	$(J_{SV}^{1,1}, J_{EV}^{1,1})$	$\cdots$	$(J_{SV}^{1,M_{EV}}, J_{EV}^{1,M_{EV}})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\pi_{SV}^{M_{SV}}$	$(J_{SV}^{M_{SV},1}, J_{EV}^{M_{SV},1})$	$\cdots$	$(J_{SV}^{M_{SV},M_{EV}}, J_{EV}^{M_{SV},M_{EV}})$

of the matrix game. As previously mentioned in Section I, there are two recognized types of equilibria in the context of autonomous driving.

Definition 1. *(Pure-strategy Nash equilibrium). A pure-strategy Nash equilibrium is a set of players' actions, $\{\pi_i^*\}_{i \in \mathcal{N}}$ such that, for each player i, it holds that*

$$J_i(\pi_i^*, \pi_{-i}^*) \leq \min_{s_i \in \Pi_i} J_i(s_i, \pi_{-i}^*),$$

where π_{-i} represents the set of actions taken by all players except player i.

The definition above indicates that no player can reduce its cost by unilaterally changing its strategy [20].

In a Stackelberg game, the idea is that the leader can take the action first, and then the follower plays the best response action [20]. The technical definition is provided as follows.

Definition 2. *(Stackelberg equilibrium). A Stackelberg equilibrium is a pair $\{\pi_L^*, \pi_F^*(\cdot)\}$ such that*

$$\pi_L^* = \arg\min_{\pi_L \in \Pi_L} J_L(\pi_L, \pi_F^*(\pi_L)),$$

$$\pi_F^*(\pi_L) = \arg\min_{\pi_F \in \Pi_F} J_F(\pi_L, \pi_F),$$

where the subscripts, L and F, represent the leader and the follower, respectively.

The leader and follower roles are not always fixed on the road. In other words, the ego vehicle can switch between being the leader and the follower. Thus, we consider two Stackelberg equilibria: one with the ego vehicle as the leader and the other with the ego vehicle as the follower.

IV. GAME-THEORETIC BEHAVIOR PLANNER

A. Semantic-Level Action Generation

1) Actions of Ego Vehicle: In the lane-merging problem, the semantic-level decision involves selecting a gap and determining the desired lateral position. As shown in Figure 3, the ego vehicle in blue has three potential gaps to choose from. To reach the target gap, the ego vehicle needs to perform a sequence of lateral decisions. The common lateral decisions are lane changing and lane keeping. Furthermore, we introduce one additional intermediate lane, represented by the dashed blue line in Figure 3, to enable a probing decision. This allows the ego vehicle to gather information and negotiate with the surrounding vehicles. Overall, the complete lateral decision set can be defined as:

$$D^{\text{lat}} := \{\texttt{LaneKeep}, \texttt{LeftChange}, \texttt{LeftProbe}\}.$$

The semantic-level decision at decision step k is denoted by an action tuple $\pi_{\text{EV},k} := (g_k, d_k^{\text{lat}})$, where $g_k \in \{\texttt{Gap0}, \texttt{Gap1}, \texttt{Gap2}\}$ and $d_k^{\text{lat}} \in D^{\text{lat}}(g_k)$. We note that the lateral decision set is conditioned on the gap selection, which reduces the number of action tuples. For example, if the ego vehicle chooses $\texttt{Gap0}$, then the only available lateral action is to keep the current lane. Next, we construct a decision tree to enumerate all the possible decision sequences. Each node in the tree represents a decision tuple. The decision tree is rooted in the decision selected in the last planning cycle and branches out at each decision step. Due to the exponential growth of the number of decision sequences with the depth of the tree, it is necessary to prune the decision tree to limit computational complexity. By using semantic-level decisions that can be easily understood by humans, we can design some rules to prune the tree. For instance, we can restrict the number of decision changes over the planning horizon because human drivers tend to maintain their current
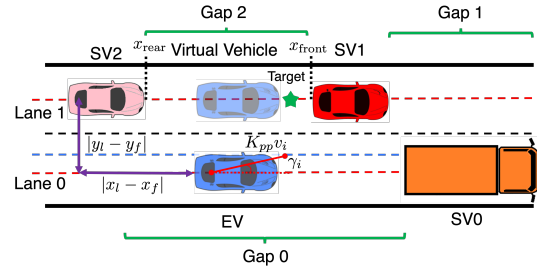

Fig. 3: Semantic-level decisions. Three gaps are available for the ego vehicle to choose from: $\texttt{Gap0}$, $\texttt{Gap1}$ and $\texttt{Gap2}$. The dashed red lines represent the centerlines of the lanes, and the dashed blue line represents the probing line. EV stands for the ego vehicle, while SV0, SV1, and SV2 represent the surrounding vehicles. $|x_l - x_f|$ and $|y_l - y_f|$ represent the longitudinal and lateral distances between the lane-changing vehicle (leader) and the interactive vehicle (follower), respectively.

driving decisions for relatively long periods of time. In addition, we can rule out certain transitions that would not be considered by normal human drivers, such as the transition from $(\texttt{Gap1}, \texttt{LeftChange})$ to $(\texttt{Gap2}, \texttt{LeftChange})$.

2) Actions of the Surrounding Vehicles: We make the following assumptions on the surrounding vehicles: (i) The surrounding vehicles maintain their lanes and have only longitudinal motion - a common assumption in prior work [8]–[10], [17], [21]; therefore, we define their longitudinal decision set as $\{\texttt{Assert}, \texttt{Yield}\}$. (ii) The surrounding vehicles maintain their decisions throughout each forward simulation. This assumption is reasonable in practice since the planner runs in a receding horizon fashion. (iii) The ego vehicle only directly interacts with at most one surrounding vehicle throughout each forward simulation. For instance, as shown in Figure 3, if the ego vehicle selects $\texttt{Gap2}$, its motion affects SV2, but not the vehicles ahead (SV0 and SV1). Therefore, we can treat all surrounding vehicles as a group, and the action set of the group is $\Pi_{\text{SV}} := \{\texttt{Assert}, \texttt{Yield}\}$.

B. Multi-vehicle Forward Simulation

1) Vehicle Dynamics: Next, we intend to generate the trajectories by simulating the motion of the vehicles from the initial states. We represent the dynamics of vehicle i as a kinematic bicycle:

$$\dot{x}_i = v_i \cos(\theta_i), \ \dot{y}_i = v_i \sin(\theta_i), \ \dot{\theta}_i = \frac{v_i}{l}\tan(\delta_i), \ \dot{v}_i = a_i,$$

where (x_i, y_i), θ_i and v_i are the position, the heading angle, and the speed, respectively; a_i and δ_i are the acceleration and the steering angle; l represents the inter-axle distance. The configuration vector is denoted as $q_i := [x_i, y_i, \theta_i]^\top$. Since we assume that the surrounding vehicles do not make lane changes, their heading and steering angles are equal to zero during the forward simulation ($\theta_i = 0$, $\delta_i = 0$). We discretize the dynamics via the Runge-Kutta 3 method.

2) Motion of the Ego Vehicle: We use two separate controllers to generate the longitudinal and lateral motion for the ego vehicle. For the longitudinal motion, we track the target longitudinal position and the desired speed via a PD

controller. One example of the target longitudinal position within the desired gap is illustrated in Figure 3. The desired gap, `Gap2`, is defined based on the positions of the front and rear vehicles, denoted as x_{front} and x_{rear}, respectively. Similar to [5], we determine the target longitudinal position and speed using a rule-based method.

As for the lateral motion, we adopt a pure pursuit controller that requires the current vehicle speed and the target line as inputs. The steering angle is computed by $\delta_i^{\text{ctrl}} = \tan^{-1}\left(\frac{2L\sin(\gamma_i)}{K_{pp}v_i}\right)$, where γ_i represents the angle between the heading direction and lookahead direction, K_{pp} is the feedback gain, and $K_{pp}v_i$ is the lookahead distance.

3) Motion of the Surrounding Vehicles: To model the behavior of the surrounding vehicles, we propose a modified intelligent driver model (IDM). Unlike the original IDM [22], which focuses solely on car following and disregards vehicles on adjacent lanes, our modified model considers lane-changing vehicles by projecting them onto their target lanes, resulting in virtual vehicles as shown in Figure 3 Subsequently, we calculate the distance between the virtual leader and the follower using the following approach:

$$d_{\text{idm}} = |x_l - x_f|e^{\kappa|y_l - y_f|}, \quad \kappa = 2\log(\beta)/w_{\text{lane}},$$

where w_{lane} is the lane width, and β is a parameter characterizing the level of willingness to yield. In fact, by adjusting the value of β, we can model different actions performed by the group of surrounding vehicles. Specifically, a large value of β indicates that the vehicle on the target lane is less likely to yield to the lane-changing vehicle because it perceives that the projection is far away. When the lateral distance between two vehicles vanishes $\left(|y_l - y_f| = 0\right)$, the virtual distance between them is equivalent to the true distance.

C. Trajectory Evaluation

After generating multi-vehicle trajectories for each action tuple, we proceed to select a specific action tuple by solving a matrix game. For constructing the cost matrix, we first introduce the cost function J_i of vehicle i, which is typically a combination of several user-defined metrics, including safety, efficiency, comfort and navigation cost: $J_i = J_i^{\text{saf}} + J_i^{\text{eff}} + J_i^{\text{com}} + J_i^{\text{nav}}$. The value of the cost function J_i depends on the trajectories generated by the forward simulator, which are influenced by the semantic-level decision sequences of the ego vehicle (π_{EV}) and the surrounding vehicles (π_{SV}).

We calculate the safety cost by examining vehicle collisions. Here, the footprint of vehicle i is modeled as a rectangle $\mathcal{R}_i(q_i)$. If the distance between two rectangles is less than a small value $\underline{d}$, indicating a potential collision, we assign a very large penalty to the corresponding trajectory. With this in mind, we compute the safety cost as follows: $J_i^{\text{saf}}(\pi_{\text{EV}}, \pi_{\text{SV}}) := \sum_{t=0}^{T}\sum_{j=1,\,j\neq i}^{N} P(q_i(t), q_j(t))$, where T is the planning horizon and N is the number of vehicles. We design P as follows:

$$P(q_i(t), q_j(t)) := \begin{cases} w_1^{\text{saf}} & \text{if } 0 \leq d_{ij}(t) < \underline{d} \\ w_2^{\text{saf}} & \text{if } \underline{d} \leq d_{ij}(t) \leq \bar{d} \\ 0 & \text{else} \end{cases}$$

where $d_{ij}(t)$ represents the distance between $\mathcal{R}_i(q_i(t))$ and $\mathcal{R}_j(q_j(t))$, and w_1^{saf} is large than w_2^{saf}. By giving a relatively small penalty w_2^{saf} when $d_{ij}(t)$ falls within the range of $[\underline{d}, \bar{d}]$, we encourage the vehicle to keep a suitable distance from the surrounding vehicles. Next, we measure the efficiency of the trajectory by computing the sum of the squares of the differences between the vehicle speed and its desired speed: $J_i^{\text{eff}}(\pi_{\text{EV}}, \pi_{\text{SV}}) := w^{\text{eff}}\sum_{t=0}^{T}(v_i(t) - v_i^{\text{des}})^2$. For the comfort cost, we consider the change in acceleration, which is known as jerk. We use the finite difference to approximate the jerk, and subsequently define the comfort cost as follows: $J_i^{\text{com}}(\pi_{\text{EV}}, \pi_{\text{SV}}) := w^{\text{com}}\sum_{t=1}^{T}(a_i(t) - a_i(t-1))^2/\Delta t^2$. Next, we penalize the differences between the vehicle lateral position and its desired lateral position to encourage the lane-changing maneuver: $J_i^{\text{nav}}(\pi_{\text{EV}}, \pi_{\text{SV}}) := w^{\text{nav}}\sum_{t=0}^{T}(y_i(t) - y_i^{\text{des}})^2$. Additionally, we introduce an information gain metric in the cost function, inspired by [23], to motivate the ego vehicle to actively identify the surrounding vehicles' intentions.

In practice, the ego vehicle needs to estimate the cost functions of other vehicles by observing their trajectories since it cannot directly access these costs. Similar to POMDP, we account for the uncertainty in the aggregate cost of the surrounding vehicles by integrating the beliefs into the cost function. The modified aggregate cost is computed as follows: $\bar{J}_{\text{SV}}^{ij} := (1 - b(\pi_{\text{SV}}^i))J_{\text{SV}}^{ij}, \quad \sum_{i=1}^{M_{\text{SV}}} b(\pi_{\text{SV}}^i) = 1$, where b represents the belief associated with the action of the surrounding vehicles. This design can be understood as incorporating prior knowledge about the behavior of the group of surrounding vehicles into the aggregate cost. For example, if we have prior knowledge suggesting that the group is likely to yield, then we can set the corresponding belief close to 1, which reduces the modified aggregate cost. We use a Bayesian estimation algorithm [24] to recursively estimate the beliefs at the beginning of each planning cycle.

After constructing the cost matrix, we compute a Nash equilibrium for the matrix game by enumerating all possible combinations of semantic-level actions. If multiple Nash equilibria exist, we select the equilibrium with the lowest social cost. If a Nash equilibrium does not exist, we choose the Stackelberg equilibrium with the ego vehicle as the follower as a backup solution.

V. ON NASH AND STACKELBERG EQUILIBRIA

In this section, we examine the conditions for the existence of a pure-strategy Nash equilibrium and explore the relationship between the Nash and Stackelberg equilibrium. We consider a specific cost matrix where $\pi_{\text{SV}}^1 := \texttt{Assert}$ and $\pi_{\text{SV}}^2 := \texttt{Yield}$ as mentioned in Section IV-A. We call an action tuple $(\pi_{\text{SV}}, \pi_{\text{EV}})$ feasible if the corresponding multi-vehicle trajectories are free from collisions, and we assume that there always exists at least one feasible action tuple.

Next, we show the existence of a Nash equilibrium for the matrix game. To obtain the results of this section, we assume that for the group of the surrounding vehicles (SV), behaving politely incurs a higher cost, while the ego vehicle (EV) achieves a lower cost if SV shows polite behavior. This assumption is required by Propositions 1 and 2.

Proposition 1. *Assume that the inequalities $0 \leq J_{SV}^{1m} \leq J_{SV}^{2m}$ and $J_{EV}^{1m} \geq J_{EV}^{2m} \geq 0$ hold for all feasible action tuples and $m \in \{1, \ldots, M_{EV}\}$. If $b(\pi_{SV}^1) \geq 0.5$, then there exists $p \in \{1, \ldots, M_{EV}\}$ such that the action tuple (π_{SV}^1, π_{EV}^p) is a pure-strategy Nash equilibrium.*

Proof. We select p such that $J_{EV}^{1p} \leq J_{EV}^{1m}$ for all $m \in \{1, \ldots, M_{EV}\}$. Then, π_{EV}^p is the best response to π_{SV}^1. Using the assumptions in the proposition, we can conclude that $0 \leq J_{SV}^{1p} \leq J_{SV}^{2p}$. Furthermore, the inequality $(1 - b(\pi_{SV}^1))J_{SV}^{1p} \leq b(\pi_{SV}^1)J_{SV}^{2p}$ holds for $b(\pi_{SV}^1) \geq 0.5$. Therefore, π_{SV}^1 is the best response to π_{EV}^p, and (π_{SV}^1, π_{EV}^p) is a Nash equilibrium. $\qquad\square$

In the following, we establish a connection between a Nash equilibrium and a Stackelberg equilibrium.

Proposition 2. *If the action tuple (π_{SV}^2, π_{EV}^p) is a Nash equilibrium, then it is also a Stackelberg equilibrium with EV as the leader.*

Proof. As (π_{SV}^2, π_{EV}^p) is a Nash equilibrium, π_{SV}^2 is the best response to π_{EV}^p, and the inequality $J_{EV}^{2p} \leq J_{EV}^{2m}$ holds for all $m \in \{1, \ldots, M_{EV}\}$. Based on the inequality $J_{EV}^{1m} \geq J_{EV}^{2m} \geq 0$, we can conclude that $J_{EV}^{2p} \leq J_{EV}^{2m} \leq J_{EV}^{1m}$. Therefore, (π_{SV}^2, π_{EV}^p) is a Stackelberg equilibrium with EV as the leader. $\qquad\square$

VI. NUMERICAL SIMULATIONS

We consider a lane-merging scenario as introduced in Section III. The red (SV1) and pink (SV2) vehicles are considered potential interactive vehicles since the ego vehicle can influence their motion, while the orange vehicle (SV0) is a non-interactive dynamic obstacle. We set the initial beliefs on the interactive vehicles' decisions to $b_{SV1}(\pi_{SV}^1) = b_{SV2}(\pi_{SV}^1) = 0.5$. To simplify the notation, we use b_{SV1}^1 and b_{SV2}^1 to represent $b_{SV1}(\pi_{SV}^1)$ and $b_{SV2}(\pi_{SV}^1)$, respectively. We use a planning horizon of $T = 25$, a discretization step of $\Delta t = 0.2$ s, a decision time period of $\Delta h = 1$ s and a decision horizon of $H = 5$.

A. Monte Carlo simulations

We conduct open-loop Monte Carlo simulations to empirically verify the existence of a Nash equilibrium and investigate its relationship with the Stackelberg equilibrium. We first specify collision-free initial multi-vehicle states. Then, the state of the ego vehicle is perturbed by ± 10 m for the initial position, ± 5 m/s for the initial longitudinal speed. We run 500 simulations for each belief and compute three equilibria: Nash (NE), Stackelberg with the ego vehicle as the leader (SE$_{EV}$), and Stackelberg with the group of the surrounding vehicles as the leader (SE$_{SV}$). The results are presented in Figure 4a. A Nash equilibrium is found in all simulations despite no theoretical guarantee in general. As mentioned before, if there are multiple Nash equilibria, we select the one with the lowest social cost. We observe that the selected Nash equilibrium coincides with one of the Stackelberg equilibria.
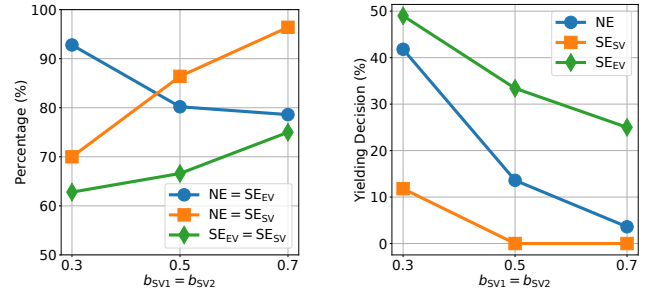
(a) Relationship between Nash and Stackelberg equilibria.

(b) Comparison of decisions in different equilibria.

Fig. 4: Results of Monte Carlo simulations. (a) illustrates the relationship between Nash and Stackelberg equilibria. (b) shows the percentage of yielding decisions made by SV in various equilibria over 500 simulations. The notations NE, SE$_{SV}$ and SE$_{EV}$ represent the Nash equilibrium, Stackelberg equilibrium with SV as the leader and Stackelberg equilibrium with EV as the leader, respectively.

Figure 4b illustrates the statistical results on SV's yielding decisions in different equilibria over 500 simulations. Specifically, SV is more likely to yield when both players employ a SE$_{EV}$. In contrast, SV is less inclined to yield when both players select a SE$_{SV}$. This implies that SV seems to behave cooperatively if the ego vehicle is the leader. On the contrary, if the ego vehicle is the follower, then it tends to show conservative behavior because the other vehicles are likely to assert. Overall, we conclude that adopting a Nash equilibrium exhibits less interactive behavior than adopting a SE$_{EV}$, but it is less conservative compared to adopting a SE$_{SV}$. In other words, it seems that a vehicle can switch between interactive and conservative behavior automatically by selecting a Nash equilibrium.

B. CARLA simulations

We conduct closed-loop simulations in the CARLA simulator to evaluate our proposed approach. The scenario depicted in Figure 3 requires the ego vehicle to merge onto the target lane as quickly as possible within a limited lane length of 100 m. To simulate a real-world scenario, we add an additional surrounding vehicle behind the pink vehicle (SV2), resulting in three surrounding vehicles on the target lane. A successful lane merging means the ego vehicle stays within 0.5 m from the center of the target lane, with a heading parallel to the lane.

We choose to control the behavior of the surrounding vehicles by the Predictive Intelligent Driver Model [25]. In this model, surrounding vehicles anticipate the ego vehicle's motion using a constant velocity model and respond when the ego vehicle comes within a certain lateral distance, which is determined by the behavior mode. The slow-moving trunk follows polite behavior, while other surrounding vehicles on the target lane exhibit selfish behavior, simulating a traffic flow with higher speeds.

We compare the proposed planner with two other baseline planners: the planner proposed in EPSILON, which selects a trajectory with the lowest cost; and a planner that selects a trajectory by seeking a Stackelberg equilibrium with the ego

vehicle as the leader. We track the trajectories generated by each planner using a lower-level PID controller.

We conduct experiments on three different planners using two initial traffic speeds. For each initial condition, we run 200 simulations to ensure statistical significance. In Table II, we observe that all the planners performed well at a low speed. However, as the traffic speed increases, our proposed game-theoretic method achieves a higher success rate compared to the baselines while requiring approximately the same merging time.

TABLE II: Evaluation in closed-loop simulation

Initial Speed	Metric	Nash	Stackelberg	EPSILON
Low ($\approx 5\,\mathrm{m/s}$)	success rate	99%	94%	98%
	collision rate	1%	6%	2%
	time to merge (s)	8.75	8.17	8.28
High ($\approx 10\,\mathrm{m/s}$)	success rate	95%	89%	70%
	collision rate	5%	11%	30%
	time to merge (s)	6.92	6.90	6.78

VII. CONCLUSION

This paper focuses on developing a game-theoretic planning algorithm for automated vehicles to perform lane changes in interactive environments. Our simulation study in CARLA demonstrates the superior performance of our proposed method compared to the state-of-the-art approaches. By comparing different equilibria via numerical simulations, we observe that selecting a Nash equilibrium allows for automatically switching between interactive and conservative driving behavior. Although forward simulations are time-consuming, they can be easily performed in parallel to reduce the computation time. In future work, we plan to implement our method in C++ and validate it on a hardware platform.

REFERENCES

[1] W. Zeng, W. Luo, S. Suo, *et al.*, "End-To-End Interpretable Neural Motion Planner," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 8652–8661.

[2] D. Chen and P. Krahenbuhl, "Learning from All Vehicles," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 17 201–17 210.

[3] M. Lauri, D. Hsu, and J. Pajarinen, "Partially Observable Markov Decision Processes in Robotics: A Survey," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 21–40, Feb. 2023.

[4] 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)*, Seattle, WA, USA: IEEE, May 2015, pp. 1670–1677.

[5] 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*, vol. 38, no. 2, pp. 1118–1138, Apr. 2022.

[6] S. L. Cleac'h, M. Schwager, and Z. Manchester, "ALGAMES: A Fast Solver for Constrained Dynamic Games," in *Robotics: Science and Systems XVI*, Jul. 2020. arXiv: 1910.09713 [cs].

[7] X. Liu, L. Peters, and J. Alonso-Mora, *Learning to Play Trajectory Games Against Opponents with Unknown Objectives*, Dec. 2022. arXiv: 2211.13779 [cs].

[8] 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," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 474–488, Jan. 2023.

[9] C. Wei, Y. He, H. Tian, and Y. Lv, "Game Theoretic Merging Behavior Control for Autonomous Vehicle at Highway On-Ramp," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 127–21 136, Nov. 2022.

[10] V. G. Lopez, F. L. Lewis, M. Liu, Y. Wan, S. Nageshrao, and D. Filev, "Game-Theoretic Lane-Changing Decision Making and Payoff Learning for Autonomous Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3609–3620, Apr. 2022.

[11] A. Zanardi, E. Mion, M. Bruschetta, S. Bolognani, A. Censi, and E. Frazzoli, "Urban Driving Games With Lexicographic Preferences and Socially Efficient Nash Equilibria," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4978–4985, Jul. 2021.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, PMLR, Oct. 2017, pp. 1–16.

[13] Z. Sunberg and M. J. Kochenderfer, "Improving Automated Driving Through POMDP Planning With Human Internal States," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20 073–20 083, Nov. 2022.

[14] P. Cai, Y. Luo, D. Hsu, and W. S. Lee, "HyP-DESPOT: A hybrid parallel algorithm for online planning under uncertainty," *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 558–573, Feb. 2021.

[15] M. Naghshvar, A. K. Sadek, and A. J. Wiggers, "Risk-averse Behavior Planning for Autonomous Driving under Uncertainty," Dec. 2018.

[16] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A Belief State Planner for Interactive Merge Maneuvers in Congested Traffic," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 1617–1624.

[17] R. Tian, L. Sun, M. Tomizuka, and D. Isele, "Anytime Game-Theoretic Planning with Active Reasoning About Humans' Latent States for Human-Centered Robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 4509–4515.

[18] K. Ji, N. Li, M. Orsag, and K. Han, "Hierarchical and game-theoretic decision-making for connected and automated vehicles in overtaking scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 150, p. 104 109, May 2023.

[19] B. Li, Y. Ouyang, L. Li, and Y. Zhang, "Autonomous Driving on Curvy Roads Without Reliance on Frenet Frame: A Cartesian-Based Trajectory Planning Method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 729–15 741, Sep. 2022.

[20] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory, 2nd Edition*. Society for Industrial and Applied Mathematics, Jan. 1998.

[21] Q. Zhang, R. Langari, H. E. Tseng, D. Filev, S. Szwabowski, and S. Coskun, "A Game Theoretic Model Predictive Controller With Aggressiveness Estimation for Mandatory Lane Change," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 75–89, Mar. 2020.

[22] M. Treiber, A. Hennecke, and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000, arXiv:cond-mat/0002177, ISSN: 1063-651X, 1095-3787.

[23] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: Leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, no. 7, pp. 1405–1426, Oct. 2018.

[24] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, Mar. 2002.

[25] B. Brito, A. Agarwal, and J. Alonso-Mora, "Learning Interaction-Aware Guidance for Trajectory Optimization in Dense Traffic Scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 808–18 821, Oct. 2022.