2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Game-Theoretic Strategy-Aware Interaction Algorithm with Validation on Real Traffic Data

Liting Sun[1,*], Mu Cai[2,*], Wei Zhan[1], and Masayoshi Tomizuka[1]

Abstract— **Interactive decision-making and motion planning are important to safety-critical autonomous agents, particularly when they interact with humans. Many different interaction strategies can be exploited by humans. For instance, they might ignore the autonomous agents, or might behave as selfish optimizers by treating the autonomous agents as opponents, or might assume themselves as leaders and the autonomous agents as followers who should take responsive actions. Different interaction strategies can lead to quite different closed-loop dynamics, and misalignment between the human's policy and the autonomous agent's belief over the policy will severely impact both safety and efficiency. Moreover, a human's interaction policy can change as interaction goes on. Hence, autonomous agents need to be aware of such uncertainties on the human policy, and integrate such information into their decision-making and motion planning algorithms. In this paper, we propose a policy-aware interaction strategy based on game theory. The goal is to allow autonomous agents to estimate humans' interactive policies and respond consequently. We validate the proposed algorithm with a roundabout scenario with real traffic data. The results show that the proposed algorithm can yield trajectories that are more similar to the ground truth than those with fixed policies. Also, we estimate how humans adjust their interaction strategies statistically based on the proposed algorithm.**

I. Introduction

Imagine that you are driving towards a narrow bridge with another car from the opposite direction. The bridge is narrow enough such that only one car can pass at a time. You and the other car (agent) are driving at similar speeds and are both away from the entrances at similar distances. In this two-agent game, suppose that you know exactly the other agent's reward function, can you accurately predict the agent's behavior and plan for the best responses? Unfortunately, the answer is no. There are multiple reasons that prevent you from accurate prediction. A major one lies in the uncertainties of the interaction strategies that the agent is taking. In this game, the other agent can take either cooperative strategies or competitive ones, or completely ignores you, i.e., not interacting at all. Those different policies can lead to quite different actions even though the reward function is exactly known. Moreover, as interaction goes on, the other agent might change the interaction policy. An inattentive driver might become attentive as you two get closer, and a driver who behaves aggressively might become conservative as your behavior is observed. Such uncertainties and time-varying strategies can make the prediction of their behaviors and the corresponding trajectory planning even more challenging.

The illustrative example above revealed one general problem in a two-player game: with known reward functions, the closed-loop dynamics of the game can have many different equilibriums due to different strategies of the players. Some very popular equilibriums include Nash equilibrium [1], Stackelberg equilibrium [2], and Pareto equilibrium [3]. Hence, to enable autonomous agents to effectively interact with humans in human-robot interaction scenarios such as driving, most of previous work has explicitly or implicitly assumed a fixed strategy of the human and optimize for the best action of the robot. For instance, Talebpour *et al* in [4] studied a lane-changing scenario assuming non-cooperative Nash strategies with V2V communication. On the other hand, in [5]–[8], the authors bypassed this issue by assuming that the human is an optimal agent which has direct access to the robot's future actions. However, such assumption is in general too strong to hold in practice. In [9] the authors proposed a nonlinear receding horizon game-theoretic planner using Nash equilibrium and tested the algorithm using racing cars in simulations. Similarly in [10], the Nash equilibrium solution is adopted to generate human-like motions. Many others used the Stackelberg strategy to model the interactions between the human and robot, such as [11]–[14]. In [13], Li *et al* adopted the k-level Stackelberg strategy to approximate humans. In [15], the authors discussed the impact of four different strategies over the interactions between a human driver and a vehicle collision avoidance controller.

We define *an interaction strategy as a solution type in a two-agent game*. Most of the work above has not explicitly considered the uncertainties of interaction strategies, and hence do not offer sufficient flexibilities for the autonomous agents to deal with time-varying strategies of humans during interaction. Moreover, no real traffic data has been utilized to evaluate the effectiveness of different interaction strategies.

Our key insight is that humans are flexible and uncertainty in terms of game strategies. Safety-critical autonomous agents that interacting with humans need to be aware of such uncertainties and time-varying strategies, and design decision-making and motion planning strategies that can adapt.

To address the aforementioned challenge, in this paper, we propose a strategy-aware interaction algorithm for

*The authors are equally contributed.

[1]Liting Sun, Wei Zhan, and Masayoshi Tomizuka are with the Department of Mechanical Engineering at University of California, Berkeley, CA 94720 USA {litingsun, wzhan, tomizuka}@berkeley.edu.

[2]Mu Cai is with the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China. im.mucai@gmail.com. The work was conducted during Mu Cai's visit to the University of California, Berkeley.

autonomous systems. We model the two-agent interaction problem as a two-player game. At each time step, based on observations, the robot agent updates its beliefs on potential interaction strategies of the human. Five different strategies are included in the strategy set to mimic different sophistication levels of humans: a Nash competitive strategy, a Pareto cooperative strategy, a Stackelberg strategy, a rule-based strategy and an inattentive strategy (ignoring). With such beliefs, a model predictive control (MPC) belief-space motion planning is performed. We validate the proposed algorithm on real traffic data, and estimate the statistical results on humans' selection of interaction strategies.

Our contributions in this work are summarized as follows:
A framework for strategy-aware interaction in games. Instead of assuming a fixed interaction strategy, we propose an interaction algorithm allowing autonomous agents to adaptively plan based on its estimate of the human's preferred solution types of the game. Such a framework can effectively deal with potential uncertainties and time-varyingness of the human's interaction strategy, and thus improve the safety and efficiency of the interaction.
Validation on real traffic data. We validate the proposed framework on real traffic data in a roundabout scenario. Through such validation, we collect the distributions of humans' interaction strategies over the specified five strategies. Moreover, results show that the proposed algorithm can achieve more human-like trajectories compared to other algorithms with fixed strategies.

II. Problem Statement

We consider the interaction between two vehicles: the ego vehicle $(\cdot)_{\text{ego}}$ and the other vehicle $(\cdot)_{\text{other}}$. We use $s{=}(s_{\text{ego}}, s_{\text{other}})$ to represent the state variable, $\hat{\gamma}$ for the predicted action, γ_{gt} for the ground truth action and $\tilde{\gamma}$ for all the possible actions. The closed-loop dynamics of the vehicles, which we assume to be fully observable [16], are given by

$$s^{t+1} = f(\gamma^t, s^t_{\text{ego}}, s^t_{\text{other}}) \tag{1}$$

where γ^t, s^t denote, respectively, the action and state of the vehicles at time t.

We assume that both agents are noisily rational optimizers. Namely, at time step t, each of them ($i{\neq}j{\in}\{\text{ego}, \text{other}\}$) is optimizing an individual finite-horizon (n steps) cost function C_i given by

$$C_i(\gamma_i^t) = \sum_{k=0}^{n-1} c(s^t, \gamma_i^{t+k}, \hat{\gamma}_j^{t+k}; \theta_i), \tag{2}$$

and then execute the first action and repeat the process at the next time step $t{+}1$. θ_i characterizes the preference of the vehicle i. We can see that the optimal solution for agent i depends on its estimate over the other agent's actions, denoted by $\hat{\gamma}_j^{t+k}$ for $k{=}0, \cdots, n{-}1$. Hence, in order to find the best $\gamma_i^t{=}\arg\max C_i(\gamma_i^t)$, agent i needs to estimate $\hat{\gamma}_j{=}[\hat{\gamma}_j^t, \cdots, \hat{\gamma}_j^{t+n-1}]$ based on its belief over agent j's interaction strategy.

In practice, γ^t, s^t are all continuous. However, to facilitate computation of the games, we discretize them. For instance, the action (acceleration) space can be discretize into a list with choices as $a = [-2, -1, 0, 1]m/s^2$.

III. Monte Carlo Tree Search under Different Game Strategies

A. Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is a heuristic-based search algorithm. It has been widely used to solve many game-theoretic problems such as the Go game in [17].

Based on the receding horizon algorithm, if the horizon is n steps, we need to build a tree with depth of $2n + 1$. Specifically, the root node represents the initial state of the two vehicles, with the odd levels for the decisions of the ego car and even levels for that of the other car. At each pair level, the two vehicles make decisions simultaneously, as you can see in Fig. 1.

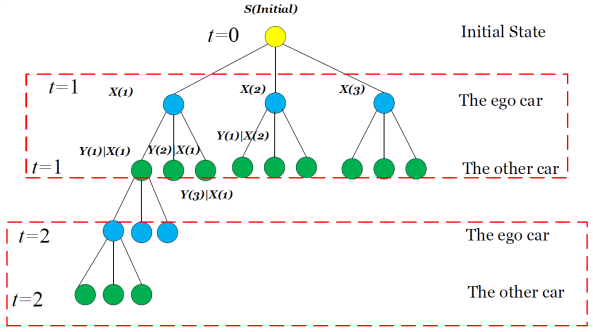


Fig. 1: A illustration of the search trees to solve a finite-horizon optimization problem: the yellow root node represents the initial state of the two vehicles, and blue nodes are the actions by the ego vehicle and green nodes are for the other vehicle. Each paired layer in the red dotted box is assumed to happen simultaneously.

Given the cost function in (2), we can design the cost function for the tree search as follows:

$$C_{MCTS}(\gamma) = C(\gamma) + \theta \cdot \sqrt{n_{\text{visit}}/log(N)} \tag{3}$$

where n_{visit} denotes the visit times of a certain children node, and N denotes that of its parent node. θ is a weighting factor balancing between the exploration and exploitation.

B. Five Interaction Strategies

In this work, we consider five different interaction strategies that the human might follow. The first three are theoretic strategies whose equilibrium types are, respectively, Nash (non-cooperative), Stackelberg (non-cooperative) and pareto (cooperative). The remaining two are incomplete-information approaches including a naive rule-based model (i.e., assuming that the human thinks that the other agent is remaining its current action through the horizon) and a ignoring policy which depicts the situation when the target human does not pay attention to the other agent. We simplify the notations as: Nash, Stackelberg, Pareto, Constant, Ignore.

1) Nash Strategy: In games, Nash equilibrium is a solution for a non-cooperative game where no player can gain more utilities by changing only their own strategy [1]. Hence, we need to find out the action sequences for both the ego and the other agent which, respectively, minimize their cost functions. The solutions should satisfy

$$\gamma_X^{t,*} = \arg\min_{\gamma_X^t} \sum_{k=0}^{n-1} c_X(s^t, \gamma_X^{t+k}, \gamma_Y^{t,*}; \theta_X), \quad (4)$$

$$\gamma_Y^{t,*} = \arg\min_{\gamma_Y^t} \sum_{k=0}^{n-1} c_Y(s^t, \gamma_X^{t,*}, \gamma_Y^{t+k}; \theta_Y), \quad (5)$$

where $(\cdot)_X$ and $(\cdot)_Y$, respectively, denote the variables for the ego vehicle and the other vehicle.

The search process with MCTS is described in Fig. 2.

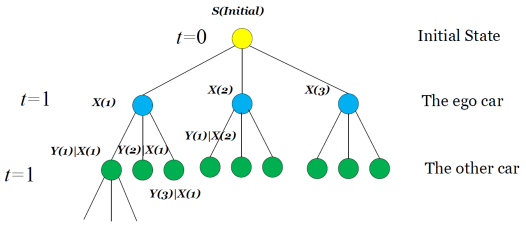


Fig. 2: The MCTS process for a Nash equilibrium

2) Stackelberg Strategy: The Stackelberg leadership model is a strategic game in which one of the players is a leader which moves first, and the other player is a follower which takes actions once he/she observes the actions from the leader [2]. In this work, we denote the ego vehicle as the leader, and the other car as the follower. With the Stackelberg model, the two players make decisions sequentially: leader first and then the follower.

More specifically, in order to choose the optimal action for the leader X at time step t, we need to find the expected response of the follower Y under a certain choice of $\gamma(X_t)$:

$$\gamma_Y^{t,*}|\gamma_X^{t,\cdots,t+n-1} = \arg\min_{\gamma_Y^{t+1}} \sum_{k=0}^{n-1} c_{follower}(\gamma_Y^{t+k}, \gamma_X^{t+k}) \quad (6)$$

Then, based on the best response of Y for each possible action of X, we can find the best decision of X as follows:

$$\gamma_X^{t,*} = \arg\min_{\gamma_X^t} \sum_{k=0}^{n-1} c_{leader}(\gamma_X^{t+k}, \gamma_Y^{t+k,*}|\gamma_X^{t+k}) \quad (7)$$

We summarize the MCTS process in Fig. 3.

3) Pareto Strategy: Pareto efficiency or pareto optimality is a state of allocation of resources from which it is impossible to reallocate so as to make any one individual or preference criterion better off without making at least one individual or preference criterion worse off. In our case, we view two players equally important, which means the target cost function should be:

$$\begin{aligned}(\gamma_X^{t,*}, \gamma_Y^{t,*}) = \arg\min_{\gamma_X^t, \gamma_Y^t} [&\sum_{k=0}^{n-1} c_X(\gamma_X^{t+k}, \gamma_Y^{t+k}, \theta_X) \\ &+ \sum_{k=0}^{n-1} c_Y(\gamma_X^{t+k}, \gamma_Y^{t+k}, \theta_Y)]\end{aligned} \quad (8)$$

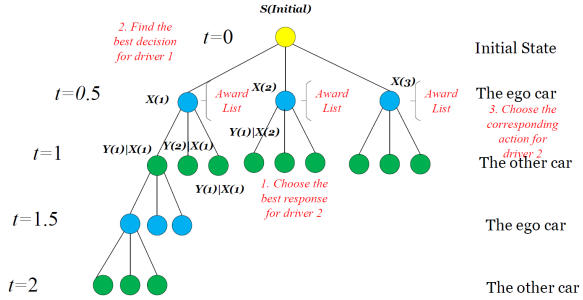


Fig. 3: The MCTS process for a Stackelberg equilibrium in a two-player game.

Hence, the depth of the tree will reduce by half, but the number of nodes in each layer will double. The MCTS process for the pareto solution is shown in Fig. 4.

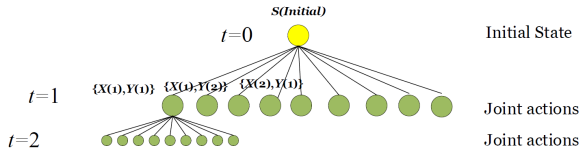


Fig. 4: The MCTS process for pareto equilibrium

4) Constant-Action Strategy: Using autonomous vehicles as an application example, the constant action we assume is that the vehicle will maintain its current speed, i.e., the acceleration is zero. Hence, with such assumption, our decision space reduces to the action space of the ego vehicle. The depth of the search tree will reduce by half and the MCTS algorithm is described in Fig. 5.

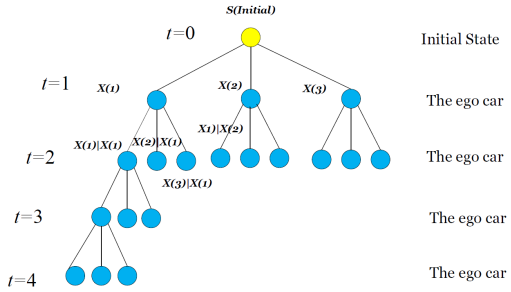


Fig. 5: The MCTS process for constant and ignoring policies

5) Ignoring Strategy: The ignoring strategy is introduced to represent the scenarios when one agent is not paying attention to the other agent that he/she is supposed to interact with. For instance, for autonomous vehicles, their detection of humans might be blocked due to sensor limitation or miss detection. We assume that under such a strategy, the ego vehicle cannot receive the state information of the other vehicle. Thus the cost function for the ego vehicle should only include terms defined on his own actions, as shown in (9). The MCTS algorithm is the same as the one for "constant-action" strategy shown in Fig. 5.

$$C_X(\gamma_X^t) = \sum_{k=0}^{n-1} c(s^t, \gamma_X^{t+k}; \theta_X) \quad (9)$$

IV. The Strategy-Aware Interaction Algorithm

To find out which strategies that humans prefer during interaction, we adopt Bayesian inference to update the probability of each strategy. Let π_i with $i{=}1,2,3,4,5$ denote the five strategies. At each time step, the probability for the i-th strategy $P(\pi_i)$ is updated recursively as follows:

$$P^t(\pi_i|\gamma^{0:t}) \propto P^{t-1}(\pi_i|\gamma^{0:t-1}) \cdot P(\gamma^t|\pi_i) \tag{10}$$

where γ^t denotes the action that a human takes at time step t. Based on the principle of Maximum Entropy, the posterior probability $P(\gamma^t|\pi_i)$ is given by

$$P^t(\gamma^t|\pi_i) = \frac{e^{-\beta Q^*(\gamma^t, s_t)}}{\sum_{\tilde{\gamma}} e^{-\beta Q_i^*(\tilde{\gamma}^t, s^t)}} \tag{11}$$

where $\tilde{\gamma}_t$ denotes all possible actions the human would take at time step t. $\beta{\geq}0$ defines how close the agents conform to the optimal strategy. As $\beta{\to}+\infty$, the agents behave more rational and vice versa. Without loss of generality, we assume $\beta{=}1$ and omit it for simplicity. This is a typical Q-value inference where Q^* denotes the cost to go given a specific action and the current state. To find Q^* under each different game policies, we run the MCTS algorithms discussed in Section III-B. Here we modify the original MSCT algorithm based on the posterior information and show how to conduct the Bayesian Inference based on game theoretic approaches in Algorithm 1.

Algorithm 1 The Bayesian Inference Algorithm

s^{ego}, s^{other}:Joint state of two vehicles
$P(\pi_i|a_t)$: The probability of adopting each policy
Covert the $X-Y$ coordinate to $l-s$ coordinate.
Compute the collision point and transfrom the origin of the $l-s$ coordinate to that point.
Extract s^{ego}, s^{other} within the interaction period
Initialize time $t=1$.
while $l_t^{ego} > 0$ and $l_t^{other} > 0$ **do**
    for $\pi_i (i = 1, ..., 5)$ **do**
        Compute posterior probability $P^t(a_t|\pi_i)$ using the cost value of children nodes using (s_t^{ego}, s_t^{other}) in Algorithm 1.
    end for
    Update the prior probability $P^t(\pi_i|a_t)$ in (10)
    $t = t+1$.
end while

Once the beliefs on policies are obtained, we integrate such information into the motion planning algorithm for the robot systems so that an expected utility under the game policy uncertainties can be maximized, as given below:

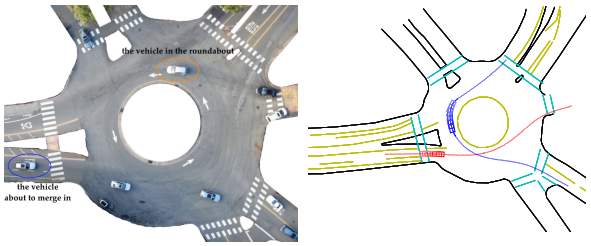
$$\gamma_{\text{ego}}^{t,*} = \arg\min_{\gamma_{\text{ego}}^t} \sum_{i=1}^{5} P(\pi_i) \sum_{k=0}^{n-1} c(\gamma_{\text{ego}}^{t+k}, \hat{\gamma}_{\text{other}}^{t+k}|\pi_i) \tag{12}$$

With such a policy-aware interaction strategy, we can design safer autonomous systems. Moreover, if the human switch policies, the robot can efficiently identify that and adapt its behaviors accordingly.

V. A Case Study

A. Real Traffic Data on a Roundabout

We use the real traffic data in a roundabout to validate the effectiveness of the proposed strategy. We consider the interactive merging at the roundabout from the INTERACTION dataset [18], [19]. As shown in Fig. 6, one ego vehicle is trying to merge into the roundabout, and the other vehicle is already in the roundabout, passing by the entrance where the first vehicle comes from. We collected 253 pairs of such interaction trajectories where each trajectory contains a sequence of the vehicle's states and actions including $x-y$ coordinates, speeds, yaw angles and accelerations. We convert the trajectories from $x-y$ coordinates to coordinates in Frenet frame $(s-d)$ based on the centerline of lane, with the origin point defined at the merging point of both paths.



(a) The roundabout map and interactive vehicles (b) An example of the extracted trajectories for interactive vehicles

Fig. 6: Illustration of the real traffic data

B. Experiment Settings

We select the planning horizon as $n{=}5$ with a time interval $\triangle t{=}0.2$s. As to the cost functions of both vehicles, to avoid significant biases, we only consider the must-have features for driving in the assumed cost functions, i.e., features for speed and collision-avoidance. For agent $i{\in}\{\text{ego}, \text{other}\}$, The cost $c(\gamma_i^t)$ at each step t is defined as

$$c(\gamma_i^t) = w_{1,i} \cdot |v_i^t - v_{des}| + w_{2,i} \cdot \varphi\left(s_i^t\right) \varphi\left(s_t^j\right) \lambda\left(s_t^i, s_t^j\right) \tag{13}$$

where $|v_i^t - v_{des}|$ quantifies the speed deviation from the desired speed v_{des}, and $\varphi\left(s_i^t\right) \varphi\left(s_t^j\right) \lambda\left(s_t^i, s_t^j\right)$ measures the collision penalty (safety-related) of two vehicles. $w_{1,i}$ and $w_{2,i}$ are, respectively, the weights for the speed and the safety term. In this work, we set $w_{i,1} = 0.05$ and $w_{i,2} = 5$. The definition of $\varphi(\cdot)$ and $\lambda(\cdot,\cdot)$ are given by:

$$\varphi(x) = \begin{cases} x, & \text{if} \quad 0 < x < K, \\ 0, & \text{else} \end{cases} \tag{14}$$

$$\lambda(x, y) = |K - |x - y|| \tag{15}$$

The hyper parameter K is a safety distance term. It is chosen based on our degree of emphasis on safety. In our experiment, we recommend $K \in [5, 10]$. The desired speed v_{des} is calculated based on the speed limit, the geometry of the path and human's acceptance range for lateral accelerations:

$$v_{des} = \text{clip}(\sqrt{\frac{1.4}{|\kappa|}}, 0, v_{limit}) \tag{16}$$

where κ denotes the curvatures of the reference curve, and v_{limit} is the speed limit. In this environment, we set it as 25mph as posted in the real map.

VI. The Results

Two studies were conducted. First, we evaluate the effectiveness of the proposed algorithm and compare it with traditional motion planning algorithms with a fixed game strategy. Second, we try to answer an motivation question - what strategies do most human take during interaction? We collected the statistical results via the Bayesian strategy inference algorithm in Algorithm 1.

A. Performance of the Strategy-Aware Algorithm

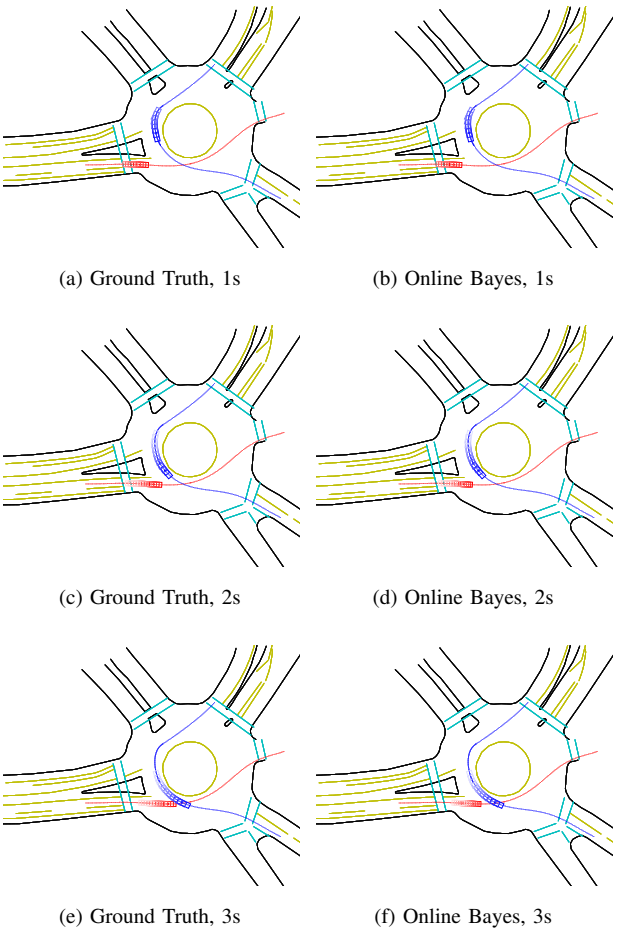
We evaluate the performance of the proposed algorithm by comparing the planned trajectories with ground-truth ones. Comparison study is conducted between the proposed algorithm and motion planning algorithms with fixed strategies. Regarding those with fixed policies, we let the ego vehicle execute exactly what the other vehicle assumes to adopt. For instance, in a motion planning with fixed Stackelberg strategy, the ego vehicle will behave as a leader and treat the other vehicle as a follower. We calculate the mean square error (MSE) of trajectories. The results are shown in Table I. We can see that the proposed algorithm (Online Bayesian) can generate trajectories that are closer to the ground-truth ones compared to those with fixed strategies. The Stackelberg strategy also achieved quite similar performance. Such results indicate that human drivers might prefer Stackelberg strategy more, and they also adapt their strategies based on observations from other interacting agents. Some illustrative examples are also given Fig. 7, where the left column shows the ground-truth data, and the right column shows the results from our algorithm at different time steps. We can see that the trajectories are quite similar. We also

TABLE I: The MSE between the ground-truth trajectories and the generated ones under different policies

Policy	MSE
Nash	$0.10042459 \pm 0.05012696$
Stackelberg	$0.09466069 \pm 0.04708792$
Pareto	$0.10011574 \pm 0.05001701$
Constant	$0.10070166 \pm 0.05012868$
Ignore	$0.12029486 \pm 0.05536152$
Online Bayes	$0.09187242 \pm 0.04990228$

calculated the minimum distance over all records using different motion planning strategies: ours (Bayes) and those with fixed policies. Results are shown in Fig. 8. We can see that the proposed algorithm performs a little bit conservative than the ground-truth trajectories, but it is safer than those with "Nash", "Pareto", "Ignore" and "Constant", and more efficient than those with "Stackelburg".

Another evaluation metric we adopted is whether the motion planning algorithms can achieve the same interaction results in terms of which vehicle passes the merging point first. If one result from a motion planning strategy differs from the ground truth, we will count as one "passing order change" (POC). The results under the proposed strategy and those with fixed strategies are shown in Fig. 9(a). We can see that the proposed strategy-aware interaction algorithm can best preserve the original passing order from the ground-truth data. Fig. 9(b) shows the results on POC with different weights between the speed feature and the safety feature in the cost functions we use. We can see that as long as the weight ratio between the two features can capture the fact that humans care more on collision, the results of using different policies in terms of POC did not change too much, particularly for the proposed approach. This means that the proposed algorithm is not very sensitive to the selection of weight ratio in the cost function.



(a) Ground Truth, 1s (b) Online Bayes, 1s

(c) Ground Truth, 2s (d) Online Bayes, 2s

(e) Ground Truth, 3s (f) Online Bayes, 3s

Fig. 7: Some illustrative examples.

B. Statistical Results of Human Drivers' Policies

The second study we conducted is to estimate the human drivers' interaction strategies using the Bayesian inference algorithm in Algorithm 2 using the real traffic data. Figure 10 shows some exemplar results where (a)-(e) are the results where the human driver each has a dominating interaction strategy, while in (f) it shows a strategy switching or drifting. Initially the human driver tends to assume that the ego vehicle is cooperative (Pareto strategy), but later on the

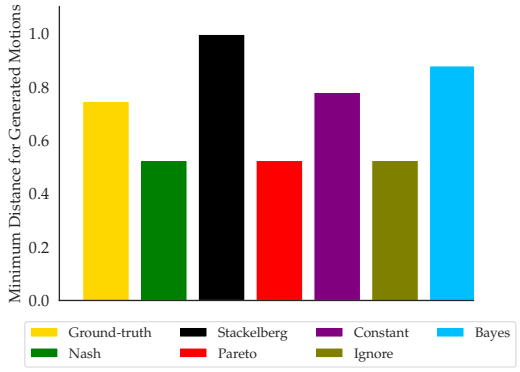


Fig. 8: The minimum distances under different policies

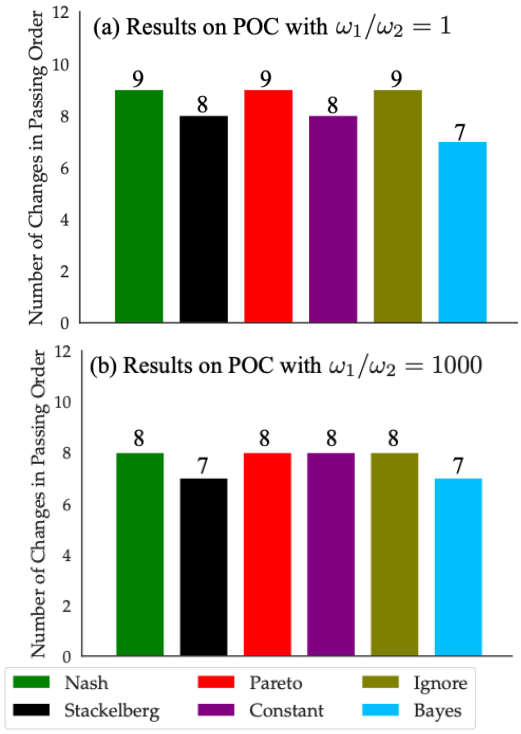


Fig. 9: Results on POC with different weight ratios

human driver maintains suspicious about several potential policies that the ego vehicle might take, although he/she tends to believe more on the non-cooperative Nash strategy.

To collect the statistical results, we run the Bayesian inference algorithm on all the 199 pairs of interactive trajectories. We calculate the following two items:

- policy switching frequency (PSF): this measure counts how many times a human switches the policies during one interaction. Such a measure can give us hints on how frequently human drivers switch policies. For instance, in the result shown in Fig. 10(a)-(e), the target human did not switch policies, while in Fig. 10(f), the human tended to switch.
- Dominance of policy (DOP): this measure quantifies how long each policy is serving as a dominant policy. As shown in Fig. 10(f), there were potentially two dominant policies - the Pareto and Nash. We also considered the duration period for each policy in this measure, i.e., the dominating policies will be weighted by their dominant periods.

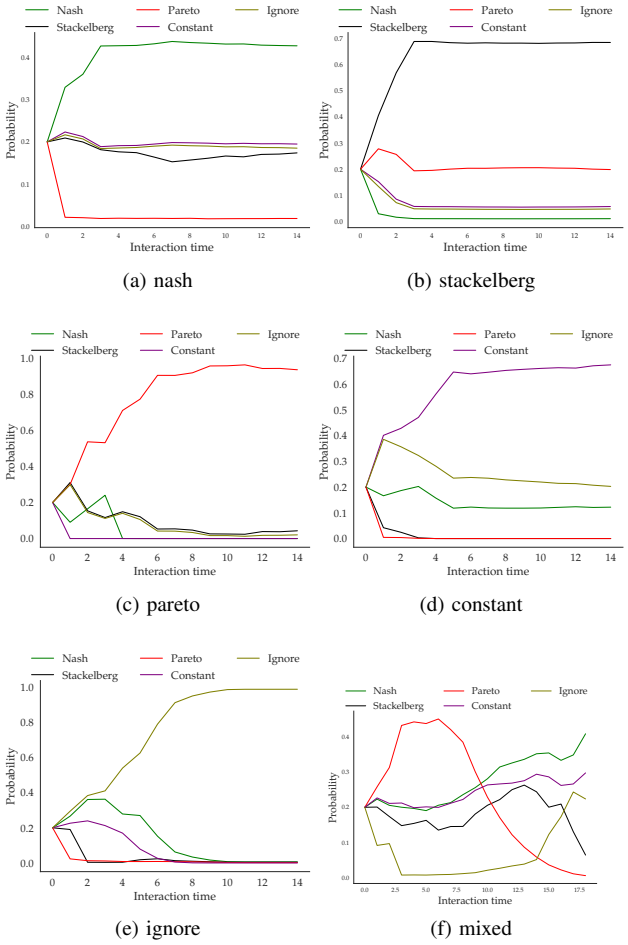


Fig. 10: Some illustrative examples of the policy inference

Figure 11 shows the results on PSF. Among the 253 pairs of interactive trajectories, in most cases, human switched only once or twice for the game strategies. There is a significant gap between three times and more. Such results can serve as important prior knowledge when we design robots' behavior. We should not let the robot switch strategies too often and should not let the robot assuming that the humans might switch strategies too often, for instance, more than three times during one interaction.

Results on the DOP are shown in Fig. 12. We can see that human drivers during interaction are not that aggressive. In most cases, particularly when two vehicles are still far away from the merging point, the human drivers would like to assume that the other driver is not paying attention to themselves and thus behave cautiously, i.e., they are running the "ignoring" strategy. When they are interacting, they tend to be more cooperative than competitive since the "Pareto" strategy dominates more than "Nash", "Stackelberg" and "Constant" policies. Such results match our observations

with courteous driving in human drivers in [7]: human tends to be cooperative and courteous to each other during interaction in most scenarios.

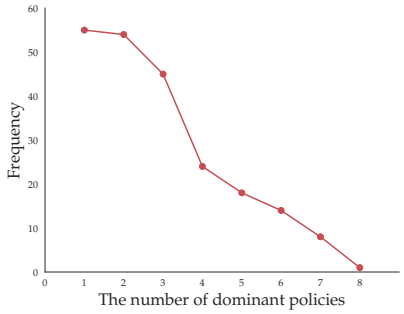


Fig. 11: Results on PSF: in most scenarios, human will not switch policies for more than three times.

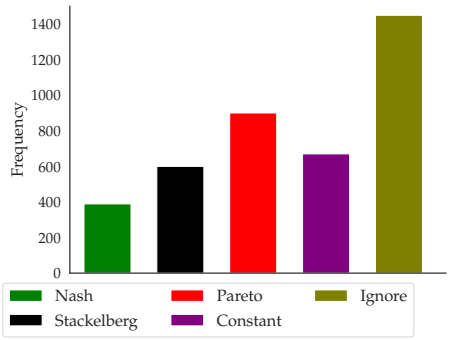


Fig. 12: Results on DOP: in most cases, human drivers are not interacting intensively, i.e., they are running the "ignoring" strategy. When they are interacting, they tend to be more cooperative than competitive since the "pareto" strategy dominates more.

VII. Conclusion

In this paper, we designed a strategy-aware interaction algorithm for safety-critical autonomous agents that interact with humans. Based on a game-theoretic setting, we designed a Bayesian inference algorithm to in-situ estimate the possible human policies against robot agents. Based on that, a strategy-aware motion planning algorithm was developed to generate safe actions in the presence of uncertain and time-varying interaction strategies. We evaluated the strategy performance via two studies with real traffic data: one on the comparison between the ground-truth and the generated trajectories, and the other one on the statistical model of human policies in driving. We found that the proposed policy-aware strategy can achieve more human-like trajectories. Humans were also found to be cooperative in most scenarios without switching policies frequently.

The work in this paper is one step further towards human-like behavior design for autonomous systems. The work can be further extended. For instance, more statistical results on different interaction scenarios can be obtained to enhance prior knowledge on human behavior for the research community. We can also explore strategies to integrate the policy inference to reward learning so that reward functions can be more accurately recovered when strategy uncertainties exist.

References

[1] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.

[2] M. Simaan and J. B. Cruz, "On the stackelberg strategy in nonzero-sum games," *Journal of Optimization Theory and Applications*, vol. 11, no. 5, pp. 533–555, 1973.

[3] S. Wang, "Existence of a pareto equilibrium," *Journal of Optimization Theory and Applications*, vol. 79, no. 2, pp. 373–384, 1993.

[4] A. Talebpour, H. S. Mahmassani, and S. H. Hamdar, "Modeling lane-changing behavior in a connected environment: A game theory approach," *Transportation Research Part C: Emerging Technologies*, vol. 59, pp. 216–232, 2015.

[5] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and Systems*, vol. 2. Ann Arbor, MI, USA, 2016.

[6] 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, 2018.

[7] L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan, "Courteous autonomous cars," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 663–670.

[8] L. Sun, W. Zhan, Y. Hu, and M. Tomizuka, "Interpretable modelling of driving behaviors in interactive driving scenarios based on cumulative prospect theory," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4329–4335.

[9] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game theoretic planning for self-driving cars in competitive scenarios," in *Robotics: Science & Systems*, 2019.

[10] A. Turnwald and D. Wollherr, "Human-like motion planning based on game theoretic decision making," *International Journal of Social Robotics*, vol. 11, no. 1, pp. 151–170, 2019.

[11] J. H. Yoo and R. Langari, "A stackelberg game theoretic driver model for merging," in *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection, 2013.

[12] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9590–9596.

[13] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 321–326.

[14] H. Yu, H. E. Tseng, and R. Langari, "A human-like game theory-based controller for automatic lane changing," *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 140–158, 2018.

[15] X. Na and D. J. Cole, "Game-theoretic modeling of the steering interaction between a human driver and a vehicle collision avoidance controller," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 25–38, 2014.

[16] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. Shankar Sastry, and A. D. Dragan, "Hierarchical Game-Theoretic Planning for Autonomous Vehicles," *arXiv e-prints*, 10 2018.

[17] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[18] W. Zhan, L. Sun, D. Wang, Y. Jin, and M. Tomizuka, "Constructing a highly interactive vehicle motion dataset," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6415–6420.

[19] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.