Anytime Game-Theoretic Planning with Active Information Gathering on Humans' Latent States for Human-Centered Robots

Ran Tian, Liting Sun, Masayoshi Tomizuka, and David Isele

***Abstract*— A human-centered robot needs to reason about the cognitive limitations and potential irrationality of its human partner to achieve seamless interactions. This paper proposes a novel anytime game-theoretic planning framework that integrates iterative reasoning models, partially observable Markov decision process, and Monte-Carlo belief tree search for robot behavioral planning. Our planner equips a robot with the ability to reason about its human partner's latent cognitive states (bounded intelligence and irrationality) and enables the robot to actively learn these latent states to better maximize its utility. Furthermore, our planner handles safety explicitly by enforcing chance constraints. We validate our approach in an autonomous driving domain where our behavioral planner and a low-level motion controller hierarchically control an autonomous car to negotiate traffic merges. Simulations and user studies are conducted to show our planner's effectiveness.**

I. Introduction

Human-centered robots operate in close proximity to humans. When designing planning algorithms for human-centered robots, it is critical for the robot to reason about the mutual influence between itself and human actors. Such a mutual dependency can be formulated as a general-sum game, in which a standard approach is to assume that each agent is a perfectly rational, expected utility maximizer, who simultaneously responds optimally to all the others (i.e., operates under equilibrium strategies) [1], [2]. However, experimental studies [3]–[5] suggest that human behaviors often systemically deviate from equilibrium behaviors due to their latent cognitive states: *bounded intelligence* (cognitive limitation) and *irrationality* (tendency to make errors). Therefore, a robot must account for its human partner's cognitive states for seamless and safe interactions.

Recent works exploited the follower model [6]–[9], and the level-k thinking [10]–[13] to equip robots with the ability to reason about non-equilibrium behaviors. These planners either assign humans' latent states a priori, omitting humans' distinct cognitive characteristics, or passively adapt to the humans' latent states, sacrificing the benefits from actively learning the latent states when planning (refer to Sec. II).

We propose an anytime game-theoretic planning framework for human-centered robots, which, to our knowledge, is the first to integrate iterative reasoning models, a partially observable Markov decision process (POMDP), and Monte-Carlo belief tree search. Drawing inspiration from behavioral game theory, we model humans' intelligence levels and degrees of rationality as their latent cognitive states, capturing their heterogeneous cognitive limitations and tendencies to make errors. Rather than passively adapting to humans' latent states when planning, our approach enables the robot to actively and safely learn the latent states to better achieve its goal. Our key insight is: *Human-centered robots can exploit the mutual influence in interactions to design actions that reveal their human partners' cognitive limitations and degrees of rationality. By actively gathering information on these latent states, robots can achieve more effective planning.* Overall, we make the following contributions:

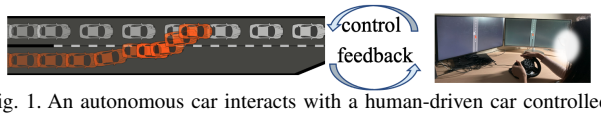

Fig. 1. An autonomous car interacts with a human-driven car controlled by a user through a racing wheel and a pedal in a forced merging scenario.

Formulation of planning with humans' bounded intelligence and irrationality. We formalize the robot planning problem in a human-robot team as a POMDP. We exploit the quantal level-k model from behavioral game theory to model the human's bounded intelligence and irrationality, which are modeled as latent states in the POMDP. In addition, a value iteration algorithm is proposed to construct closed-loop human behavioral models under various intelligence levels and degrees of rationality, and such models are further embedded in the POMDP through belief dynamics.

An active anytime game-theoretic planner. The POMDP is represented as a stochastic optimal control problem with *safety chance constraints* and solved by an open-loop Monte-Carlo belief tree search algorithm in an anytime manner. Coupled with explicit realization of active information gathering on the human's latent states and tailored terminal value functions, our planner enables the robot to safely and adaptively balance between exploration (learning the human's latent states) and exploitation (maximizing utility).

Application of our framework to autonomous driving. The proposed behavioral planner and a low-level motion control layer are hierarchically connected to achieve feedback control for an autonomous car represented by a high-fidelity model. Simulations and user studies are conducted to show the effectiveness of our planner compared to baselines.

II. Related Work

Game-theoretic planning for human-centered robots. Our work is related to [6]–[14]. In [6]–[9], the robot exploits the Stackelberg game [15] and models its human partner as a pure/uncertain *follower* who accommodates the robot's planned actions. The follower model is homogeneous (the human is always the follower w.r.t. the robot), thus the robot may behave poorly when the human does not behave like a follower. In [10]–[13], the robot exploits the level-k model [16] to model the human as an agent who reasons under various intelligence levels. While the level-k model

Ran Tian, Liting Sun, and Masayoshi Tomizuka are with the UC Berkeley. ({rantian,liting,tomizuka}@berkeley.edu). David Isele is with Honda Research Institute. (disele@honda-ri.com).

is heterogeneous, it assumes that humans with lower-levels of intelligence best respond to higher-level humans, omitting humans' potential irrationalities. Furthermore, the planners in [10]–[12] *passively* adapt to the human's intelligence level, leading to less effective plans. The approach in [6] allows the robot to *actively* "probe" the human's latent states, but the underlying human model is a pure follower model. [14] enables the robot to infer humans' leading/following roles using a graph, which is different from the iterative reasoning model considered in this paper. In addition, the safety of the planner is not explicitly enforced in [6], [14]. *Here, we embrace that humans have heterogeneous intelligence levels and degrees of rationality as their latent states. Our planner actively learns these latent states to better maximize utility, and handles safety explicitly by enforcing chance constraints.*

Solution methods for POMDP. A POMDP is a framework for planning under uncertainty. Various approaches have been proposed to approximate POMDPs, including point-based methods [17]–[19], open-loop strategies [20]–[23], and Monte-Carlo tree search [24], [25]. Partially observable Monte-Carlo planning (POMCP) [24] performs well, though building a closed-loop search tree in large games is computationally expensive. Open-loop strategies condition action selection on previous action sequences. They use a much smaller search space by sacrificing the ability of active information gathering, and achieve competitive performance compared to closed-loop planning under computational constraints [21]. *Our approach combines the strengths from POMCP and open-loop strategies, achieving real-time active game-theoretic planning.*

III. Problem formulation

Human-robot team formalization. We formalize the human-robot team as a two-player dynamic game represented by the tuple $\mathcal{G} =< \mathcal{P}, \tilde{\mathcal{S}}, \mathcal{A}, f, r_{\mathcal{R}}, r_{\mathcal{H}} >$, where $\mathcal{P} = \{\mathcal{R}, \mathcal{H}\}$ represents the two players with $\mathcal{R}$ denoting the robot and $\mathcal{H}$ denoting the human; $\tilde{\mathcal{S}} = \tilde{\mathcal{S}}_{\mathcal{R}} \times \tilde{\mathcal{S}}_{\mathcal{H}}$ and $\mathcal{A} = \mathcal{A}_{\mathcal{R}} \times \mathcal{A}_{\mathcal{H}}$ are, respectively, the joint fully-observable state and action spaces of the two agents; the function f governs the evolution of the joint fully-observable state and is defined by the following dynamic model: $\tilde{s}_{t+1} = f(\tilde{s}_t, a_t^{\mathcal{R}}, a_t^{\mathcal{H}})$ ($\tilde{s}_t \in \tilde{\mathcal{S}}$, $a_t^{\mathcal{R}} \in \mathcal{A}_{\mathcal{R}}$, $a_t^{\mathcal{H}} \in \mathcal{A}_{\mathcal{H}}$); $r_{(\cdot)} : \tilde{\mathcal{S}} \to \mathbb{R}$ is the reward function of an agent.

Robot planning as a POMDP. We consider planning from the robot's perspective. The fully-observable state $\tilde{s}$ of the human-robot team represents measurable variables (e.g., position, speed, etc.). In addition, the human also has latent states (e.g., preference, trust, cognitive limitation, etc.) that characterize his/her cognition and reasoning; such latent states cannot be observed directly, and therefore must be inferred from interactions. We let $\theta \in \Theta$ denote the human's latent states, and we consider an augmented state space $\mathcal{S} = \tilde{\mathcal{S}} \times \Theta$. As the robot's knowledge about the augmented state $s \in \mathcal{S}$ is incomplete, it maintains a belief distribution over $\mathcal{S}$ at each discrete time step t, namely, the robot maintains the belief state $b_t := [\mathbb{P}(s_t = s_1), \ldots, \mathbb{P}(s_t = s_{|\mathcal{S}|})]^\intercal$. We formulate the robot planning problem as a POMDP defined by the tuple $< \mathcal{G}, \mathcal{S}, \mathcal{B}, \Omega, \mathcal{Z}, \rho, r'_{\mathcal{R}}, \mathbb{O}_{\text{safe}} >$, where $\mathcal{G}$ denotes the dynamic game model defined above; $\mathcal{S}$ is the augmented state space; $\mathcal{B}$ represents the space of probability distributions over $\mathcal{S}$ ($b_t \in \mathcal{B}$); Ω is the finite observation space; $\mathcal{Z} : \Omega \times \mathcal{S} \to [0, 1]$ is a probability function specifying the probability of receiving an observation in a state; the belief dynamics function $\rho : \mathcal{B} \times \mathcal{A}_{\mathcal{R}} \times \Omega \to \mathcal{B}$ governs the belief state evolution and is defined as $b_{t+1} = \rho(b_t, a_t^{\mathcal{R}}, o_{t+1})$. Given an initial belief state b_t, the robot executes the action $a_t^{\mathcal{R}}$, receives the observation o_{t+1} at $t+1$, and updates its belief accordingly. $r'_{\mathcal{R}} : \mathcal{B} \times \mathcal{A}_{\mathcal{R}} \to \mathbb{R}$ denotes the reward function of the robot in belief space (defined in Sec. V-A); $\mathbb{O}_{\text{safe}} \subseteq \tilde{\mathcal{S}}$ represents the set of safe states of the robot.

We let $\pi_{\mathcal{R}} : \mathcal{B} \to \mathcal{A}_{\mathcal{R}}$ denote a robot's deterministic policy. Given belief state b_t, the robot maximizes its value:

$$\pi^*_{\mathcal{R}} = \arg\max_{\pi} V^{\pi}_{\mathcal{R}}(b_t), \tag{1}$$

where $V^{\pi}_{\mathcal{R}}(b_t) = \mathbb{E}_{\mathcal{Z}}\left[\sum_{\tau=0}^{\infty} \gamma^{\tau} r'_{\mathcal{R}}(b_{t+\tau}, a_{t+\tau}) \middle| a_{t+\tau} = \pi(b_{t+\tau})\right]$ is the value function representing the robot's expected return starting from b_t, subject to its policy and the belief dynamics function. Note the robot needs to reason about both its own actions and the human's responses due to the mutual dependence. The POMDP formulation allows us to condition the robot behaviors on the inferred latent states.

IV. Human latent states modeling

A. Iterative Human Reasoning Model

When studying interactions in games, players are commonly assumed to adopt the Nash equilibrium solution: each player has unlimited computational resources and responds optimally to the others. In real life however, humans are known to act irrationally and have cognitive limitations. The iterative human reasoning models from behavioral game theory have been proven to show better performance in characterizing humans' reasoning capabilities in simultaneous games [5]. Examples of iterative human reasoning models include: the level-k model [16], the cognitive hierarchy model [26], and the quantal level-k model [27]. All these models aim to capture humans' cognitive limitations and share a common feature: they model humans as agents with heterogeneous *bounds* on their reasoning processes, i.e., human agents can only perform a finite number of iterations of reasoning, and such an intelligence bound is referred as the *intelligence level*. Among the various integrative reasoning models, the quantal level-k model is the state-of-the-art [28].

B. Human Quantal Level-k Reasoning Model

Quantal best response and rationality coefficient. One of the key components of the quantal level-k (ql-k) model is quantal best response. The notion behind quantal best response is that human players are more likely to select actions with higher expected future rewards [29]. Formally, we define the quantal best response function as follows: let $Q^i(\tilde{s}, a^i | a^{-i})$ denote agent i's expected total reward ($i \in \mathcal{P}$) when executing a^i in $\tilde{s}$ against an action a^{-i} from his/her opponent $-i$. Then a quantal best response by agent i to agent $-i$ is a mixed policy:

$$\pi^i(\tilde{s}, a^i | a^{-i}) = \frac{\exp\left(\lambda^i Q^i(\tilde{s}, a | a^{-i})\right)}{\sum_{a' \in \tilde{\mathcal{A}}_i} \exp\left(\lambda^i Q^i(\tilde{s}, a' | a^{-i})\right)}, \tag{2}$$

where $\lambda^i \in (0, 1]$ is the *rationality coefficient* that controls the degree of agent i conforming to optimal behaviors. In general, the larger the λ is, the more rational the human is.

Human quantal level-k policies. In the ql-k model, the iterative reasoning process starts from ql-0 agents who are non-strategic reasoners. Then, a ql-k agent, $k \in \mathbb{N}^+$, assumes the other agents are ql-$(k-1)$ agents, predicts their ql-$(k-1)$ policies, and quantally best responds to the predicted ql-$(k-1)$ policies. On the basis of ql-0 policies, the ql-k policies are defined for every $i \in \mathcal{P}$, for every $\lambda \in \Lambda$, and for every $k = 1, \ldots, k_{\max}$ through a sequential and iterative process. Specifically, given an initial state $\tilde{s}_t \in \tilde{\mathcal{S}}$, a ql-$k$ agent i maximizes the following objective: $\max_{\pi^{i,k,\lambda^i}} V^{i,k}(\tilde{s}_t)$, where $V^{i,k}(\tilde{s}_t) = \mathbb{E}_{\pi^{*,-i,k-1,\lambda^{-i}}}\left[\sum_{\tau=0}^{\infty} \gamma^\tau r_i(\tilde{s}_{t+\tau})\right]$ is the ql-k value function of agent i and $\pi^{*,-i,k-1,\lambda^{-i}} : \tilde{\mathcal{S}} \times \mathcal{A}_{-i} \to [0,1]$ is the predicted ql-$(k-1)$ policy of agent $-i$. The optimal value function satisfies the following Bellman equation: $V^{*,i,k}(\tilde{s}) = \mathcal{B}\, V^{*,i,k}(\tilde{s}) = \max_{a^i \in \mathcal{A}_i} \mathbb{E}_{\pi^{*,-i,k-1,\lambda^{-i}}}\Big[r_i(\tilde{s}') + \gamma V^{*,i,k}(\tilde{s}') \big| \tilde{s}' = f(\tilde{s}, a^i, a^{-i}), a^{-i} \sim \pi^{*,-i,k-1,\lambda^{-i}}\Big]$, and can be determined via value iteration. Then, we define the Q-value function as:

$$Q^{*,i,k}(\tilde{s}, a^i) = \mathbb{E}_{\pi^{*,-i,k-1,\lambda^{-i}}}\Big[r_i(\tilde{s}) + \gamma V^{*,i,k}(\tilde{s}')\Big], \quad (3)$$

and (2) is adopted to define agent i's ql-k policy. We note that when agent i predicts its opponent's ql-$(k-1)$ policy, it assumes that its opponent's rationality coefficients is also λ^i (i.e., $\lambda^{-i} = \lambda^i$) and forms its policy based on $\pi^{*,-i,k-1,\lambda^i}$.

We summarize the algorithm that computes the ql-k policies of the agents in $\mathcal{G}$ in Alg. 1. Note that: 1) Alg. 1 exploits a closed-loop feedback information structure to compute game-theoretic policies, as opposed to the open-loop information structure used in [11], [30]; 2) the algorithm extends to multi-player dynamic games straightforwardly.

Algorithm 1: Quantal level-k dynamic programming

Input: The highest intelligence level $k_{\max}$, a set of rationality coefficients Λ, and the level-0 model $\pi^{i,0}$, $i \in \mathcal{P}$.
for $k = 1 : k_{max}$ **do**
  for $(i, \lambda) \in \mathcal{P} \times \Lambda$ **do**
    while $V^{i,k}$ *not converged* **do**
      for $\tilde{s} \in \tilde{\mathcal{S}}$ **do** $V^{i,k}(\tilde{s}) \leftarrow \mathcal{B}V^{i,k}(\tilde{s})$;
    for $(\tilde{s}, a^i) \in \tilde{\mathcal{S}} \times \mathcal{A}_i$ **do**
      Compute
      $$\pi^{*,i,k,\lambda}(\tilde{s}, a^i) = \frac{\exp\big(\lambda Q^{*,i,k}(\tilde{s}, a^i)\big)}{\sum_{a' \in \mathcal{A}_i} \exp\big(\lambda Q^{*,i,k}(\tilde{s}, a')\big)}$$
      using (3);
Return $\{\pi^{*,i,k,\lambda}\}$, $i \in \mathcal{P}$, $k = 1, \ldots, k_{\max}$, and $\lambda \in \Lambda$.

Summary. We model the human's latent states as his/her intelligence level and rationality coefficient, i.e., $\theta = (k, \lambda)$ and use Alg. 1 to compute the policies/value functions of the human and the robot as ql-k agents, which are exploited to solve the POMDP in Sec. V.

V. Anytime Active Game-Theoretic Planning

In this section, we embed the human ql-k model in the POMDP and present our anytime game-theoretic planner.

A. Embed the Human Behavioral Model in Robot Planning

Observation function. We define an observation made by the robot as $o := \tilde{s}$, i.e., the robot can measure the joint physical state $\tilde{s}$. Then the observation function is defined as: $\mathcal{Z}(o', s) = \mathbb{I}(o' = \tilde{s})$, where $\tilde{s}$ is the joint physical state in s, and $\mathbb{I}(\cdot)$ is an indicator function, taking 1 if the event $(\cdot)$ is true and taking 0 otherwise.

Prior belief state. We define the probability of arriving to state $s' = (\tilde{s}', \theta') \in \mathcal{S}$ from state $s = (\tilde{s}, \theta) \in \mathcal{S}$ after executing $a \in \mathcal{A}_{\mathcal{R}}$ as:

$$\mathcal{T}(s, a, s') := \mathbb{P}(s_{t+1} = s' | s_t = s, a_t^{\mathcal{R}} = a) \quad (4)$$
$$= \sum_{a' \in \mathcal{A}_{\mathcal{H}}} \mathbb{I}\big(\tilde{s}' = f(\tilde{s}, a, a')\big) \mathbb{P}(\theta_{t+1} = \theta' | \theta_t = \theta, \tilde{s}_t = \tilde{s}, \bar{\sigma}) \pi^{\mathcal{H},k,\lambda}(\tilde{s}, a'),$$

where f is the dynamics function described in Sec. III, $\mathbb{P}(\theta_{t+1} | \theta_t, \tilde{s}_t, \bar{\sigma})$ represents an explicit probabilistic model that governs the dynamics of the latent states ($\bar{\sigma}$ denotes the model parameters), and $\pi^{\mathcal{H},k,\lambda}$ denotes the human's ql-k policy with rationality coefficient λ (recall $\theta = (k, \lambda)$). Then, we can define a prior belief state prediction function that predicts the future belief state without accounting for the possible observations: $\tilde{b}_{t+1} = \tilde{\rho}(b_t, a)$, where each element in $\tilde{b}_{t+1}$ is computed following $\tilde{b}_{t+1}(s') = \sum_{s \in \mathcal{S}} \mathcal{T}(s, a, s') b_t(s)$. Then, the robot's reward function in belief space can be defined as: $r'_{\mathcal{R}}(b_t, a_t) = \sum_{\tilde{s}'} r_{\mathcal{R}}(\tilde{s}') \mathbb{P}(\tilde{s}_{t+1} = \tilde{s}' | \tilde{\rho}(b_t, a_t))$.

Prior probability of future observations. With the observation function and the prior belief prediction function defined, given an initial belief state b, we can predict the probabilities of the robot's future observations after executing an action $a \in \mathcal{A}_{\mathcal{R}}$ following:

$$\mathcal{O}(o, b, a) := \mathbb{P}(o_{t+1} = o | b_t = b, a_t^{\mathcal{R}} = a) = \sum_{s' \in \mathcal{S}} \mathcal{Z}(o, s') \tilde{b}_{t+1}(s'). \quad (5)$$

Posterior belief update. After executing an action and receiving an observation, the robot can update its posterior belief state through the belief dynamics equation $b_{t+1} = \rho(b_t, a_t^{\mathcal{R}}, o_{t+1})$. More specifically, each element in b_{t+1} can be computed using the Bayesian inference equation [31]:

$$b_{t+1}(s') = \frac{\mathcal{Z}(o_{t+1}, s') \tilde{b}_{t+1}(s')}{\mathcal{O}(o_{t+1}, b_t, a_t^{\mathcal{R}})}, \quad \forall s' \in \mathcal{S}. \quad (6)$$

Summary. The human's behavioral model is embedded into the belief transition (4). With (6), the robot infers the human's latent states, and in turn uses the inference result to predict belief evolution via the belief prediction function $\tilde{\rho}$.

B. Robot Planning Algorithm

Closed-loop policy. In stochastic environments, the robot's actions have *dual effect* [32], [33]: effect on *controlling the state* and effect on *controlling the uncertainty*. Namely, the actions can help "actively learn" the latent states for the benefits of the future. [34] shows that only the closed-loop policy has the capability of *active learning* as it optimally balances between uncertainty reduction and maximizing utility. Solving (1) via stochastic dynamic programming to obtain a closed-loop policy is computationally intractable [35]. To achieve real-time computation, we exploit open-loop strategies, but compensate for the loss of active learning.

Open-loop feedback strategy. We write the belief space planning problem as a stochastic model predictive control problem. In contrast to solving for the optimal policy in (1), the robot solves for an optimal action sequence at each t:

$$\mathbf{a}_t^{\mathcal{R}} = \arg\max_{\mathbf{a}} \mathbb{E}_{\mathcal{Z}}\left[\hat{V}(b_{t+T}) + \sum_{\tau=0}^{T-1} r'_{\mathcal{R}}(b_{t+\tau}, a_{t+\tau})\right], \quad (7)$$

where $\mathbf{a} = \{a_{t+0,\ldots,a_{t+T-1}}\}$ is a planned action sequence and $\hat{V}(b_{t+T})$ denotes the *terminal value* of the predicted belief state b_{t+T}. The robot plans in a feed-back manner by applying the first action in $\mathbf{a}_t^{\mathcal{R}}$ and re-planing at the next time step. As opposed to the closed-loop policy, (7) fixes

the planned action sequence ahead and omits the benefits that can be propagated back from the future observations. Consequently, (7) only *passively* learns the latent states and yields conservative actions. Hence, explicit methods can be exploited to realize active learning of the latent states [36].

Active learning of the latent states. We exploit the Shannon entropy [37] to measure the estimation uncertainty of a belief state, and augment the robot's reward function with the expected information gain:

$$\mathcal{I}(b,a) = H(b) - \sum_{o} \mathcal{O}(o,b,a) H\big(\rho(b,a,o)\big), \tag{8}$$

where $H(b) = -\sum_{s\in\mathcal{S}} b(s)\log b(s)$, and $\mathcal{O}$ is the observation prediction function defined in (5). In general, the higher the $\mathcal{I}(b,a)$ is, the more expected information the robot obtains about the human's latent states if the robot executes a in b.

Safety chance constraint. We impose constraints on the safety of a plan. Specifically, the robot solves for:

$$\mathbf{a}_t^{\mathcal{R}} = \arg\max_{\mathbf{a}} \mathbb{E}_{\mathcal{Z}}\left[\hat{V}(b_{t+T}) + \sum_{\tau=0}^{T-1} \tilde{r}_{\mathcal{R}}(b_{t+\tau}, a_{t+\tau})\right], \tag{9a}$$

$$\tilde{r}_{\mathcal{R}}(b,a) = r'_{\mathcal{R}}(b,a) + \eta\mathcal{I}(b,a), \tag{9b}$$

$$\text{s.t.} \quad \mathbb{P}(\tilde{s}_{t+\tau} \in \mathbb{O}_{\text{safe}} | \mathbf{a}, b_t) \geq 1 - \Delta_\tau, \sum_{\tau=1}^{T} \Delta_\tau \leq \Delta, \tag{9c}$$

where $\tilde{r}_{\mathcal{R}}(b,a)$ is the robot's augmented reward function, with $\eta \propto H(b)$ being an adaptive term. The constraint (9c) requires that the probability that the predicted state $\tilde{s}_{t+\tau}$ is in the safe set $\mathbb{O}_{\text{safe}}$ is larger than $1-\Delta_\tau$ for all steps over the planning horizon, with Δ_τ being a design parameter. The overall safety is bounded by Δ via risk allocation [38], [39].

Open-loop Monte-Carlo belief tree search. POMCP [24] combines Monte-Carlo simulation and game-tree search. Building upon POMCP, we propose an algorithm (Alg. 2) to solve (9) in an anytime manner. Our algorithm differs from POMCP in the following ways: 1) a search node, $v = \big\langle \mathbf{a}, V(\mathbf{a}), N(\mathbf{a})\big\rangle$, stores an action sequence $\mathbf{a}$ and its associated statistics: $V(\mathbf{a})$ is the mean return of all simulations that execute $\mathbf{a}$, and $N(\mathbf{a})$ counts the number of times that $\mathbf{a}$ has been visited; 2) leaf node expansions must enforce the safety chance constraint (line 14); 3) terminal values are estimated using the pre-computed ql-k values (line 11); 4) the active information gathering on latent states is explicitly realized via the augmented reward function $\tilde{r}_{\mathcal{R}}$.

Benefits of Algorithm 2. A search node only stores an action sequence since the open-loop optimization (9) searches for an action sequence. In contrast to POMCP, in which a node stores a history of actions and observations, the search space in Alg. 2 is significantly reduced. Hence Alg. 2 can run in real-time under computational constraints. Quantal level-k reasoning (Sec. IV) is exploited to estimate the terminal value when the maximum planning depth is reached (line 11). Specifically, the terminal belief state b_{t+T} is used to determine the human's intelligence level distribution $\mathbb{P}(k^{\mathcal{H}}|b_{t+T})$, then, we assume the robot behaves as a ql-$(k^{\mathcal{H}}+1)$ agent (recall that a ql-$(k+1)$ agent quantally best responds to a ql-k agent) with rationality coefficient $\lambda = 1$ and estimates the terminal value using the pre-computed robot's ql-$(k^{\mathcal{H}}+1)$ value weighted by $\mathbb{P}(k^{\mathcal{H}}|b_{t+T})$. By actively learning the belief state, the planner can quickly reduce the estimation error on the terminal values and maximize its performance.

Infeasibility handling mechanism. When the root node has no safe successors, we relax the chance constraint and find the action that minimizes the degree of constraint violation.

Algorithm 2: Open-Loop Monte-Carlo Belief Tree Search

1 **Input**: POMDP model, planning horizon T, discount factor γ, exploration parameter e, and initial belief state b_0;
2 **while** *goal not reached* **do**
3 Initialize root node $v_{\text{root}} = (\emptyset, 0, 0)$;
4 **while** *not TimeOut()* **do**
5 Simulate(v_{root}, b_t, 0); $N(v_{\text{root}}.\mathbf{a})+=1$;
6 $a_t^{\mathcal{R}} = \arg\max_{a\in\mathcal{A}_{\mathcal{R}}} V(a)$;
7 Execute $a_t^{\mathcal{R}}$ and collect the new observation o_{t+1};
8 Update belief state: $b_t \leftarrow \rho(b_t, a_t, o_{t+1})$;
9 **Function** Simulate(v, b, τ)
10 **if** $\tau = T-1$ **then**
11 return $\sum_{k\in\mathcal{K}}\Big(\mathbb{P}(k^{\mathcal{H}}=k|b)\big(\sum_{\tilde{s}\in\tilde{\mathcal{S}}}\mathbb{P}(\tilde{s}|b)V^{*,\mathcal{R},k+1}(\tilde{s})\big)\Big)$
12 **if** *IsLeafNode(v)* **then**
13 **for** *all* $a \in \mathcal{A}_{\mathcal{R}}$ **do**
14 **if** *ComputeRisk*$(\tilde{\rho}(b,a)) < \Delta_\tau$ **then**
15 Append $v_{succ} = \big\langle v.\mathbf{a}.add(a), 0, 0\big\rangle$ in the tree;
16 return *Rollout*(b, τ);
17 **else**
18 $v^*_{\text{succ}} = \arg\max_{v_{\text{succ}}} V(v_{\text{succ}}.\mathbf{a}) + e\sqrt{\frac{\log(N(v.\mathbf{a}))}{N(v_{\text{succ}}.\mathbf{a})}}$;
19 $a = v^*_{\text{succ}}.\mathbf{a}.last()$; Sample observation $o \sim \mathcal{O}(o,b,a)$;
20 $\hat{V} = \tilde{r}_{\mathcal{R}}(b,a) + \gamma$Simulate($v^*_{\text{succ}}$, $\rho(b,a,o)$, $\tau+1$);
21 $N(v^*_{\text{succ}}.\mathbf{a})+=1$; $V(v^*_{\text{succ}}.\mathbf{a})+=\frac{\hat{V}-V(v^*_{\text{succ}}.\mathbf{a})}{N(v^*_{\text{succ}}.\mathbf{a})}$; Return $\hat{V}$;
22 **end Function**
23 **Function ComputeRisk**(b)
24 return $\sum_{\theta\in\Theta}\text{proj}_\theta(b)^\intercal[\mathbb{I}(\tilde{s}^1\in\mathbb{O}^c_{\text{safe}}),\ldots,\mathbb{I}(\tilde{s}^{n_{\tilde{S}}}\in\mathbb{O}^c_{\text{safe}})]^\intercal$;
25 **end Function**
26 **Function Rollout** (b, τ)
27 **if** $\tau = T-1$ **then**
28 return $\sum_{k\in\mathcal{K}}\Big(\mathbb{P}(k^{\mathcal{H}}=k|b)\big(\sum_{\tilde{s}\in\tilde{\mathcal{S}}}\mathbb{P}(\tilde{s}|b)V^{*,\mathcal{R},k+1}(\tilde{s})\big)\Big)$
29 **else**
30 $a \sim \pi_{rollout}(b)$;$o \sim \mathcal{O}(o,b,a)$;
31 return $\tilde{r}_{\mathcal{R}}(b,a) + \gamma\cdot$Rollout($\rho(b,a,o)$, $\tau+1$);
32 **end Function**

VI. Experiments

A. Implementation Details

Test domain. We use *autonomous driving* as the test domain. In particular, we consider a forced merging scenario [40], where an autonomous car must merge to an adjacent (upper) lane that is occupied by a human-driven car (Fig. 1).

Reward functions. The reward function of a car is a linear combination of features $\phi : \tilde{S} \to \mathbb{R}^{n_f}$,satisfying $r_{(\cdot)}(\tilde{s}) = \omega^\intercal \cdot \phi(\tilde{s})$. The features encode safety, comfort, progress, etc. The design of the reward function follows from Sec. II(C) in [11]. The weights $\omega \in \mathbb{R}^{n_f}$ can be recovered via an inverse learning algorithm [41]–[43]. Note that when the autonomous car runs Alg. 2, the safety feature in its reward function is deactivated since the safety of the robot is handled by the chance constraints.

Level-0 policy. Alg. 1 requires a ql-0 policy to initiate the iterative reasoning process. Similar to [11], we let a ql-0 agent be a non-strategic agent who treats others as static obstacles when making decisions. Note that this ql-0 policy is chosen specifically for the autonomous driving setting, and it can differ according to the application.

Latent state dynamics. Recall (4) that an explicit probabilistic model is required to govern the dynamics of the

latent states. In this work, we consider single-shot games, i.e., the human's latent states are assumed to be constant during interaction [10], [11]. Hence, the latent states' transition model is reduced to $\mathbb{P}(\theta_{t+1}|\theta_t, \tilde{s}_t, \bar{\sigma})=\mathbb{I}(\theta_{t+1}=\theta_t)$. In general repeated games, the transition model can be represented as a Markov chain and its parameters $\bar{\sigma}$ can be embedded in the POMDP and learned simultaneously as in [13].

High-level robot planning. Following [8], the dynamics of the human-robot team are represented as

$$[\dot{x}_{\mathcal{R}} \quad \dot{y}_{\mathcal{R}} \quad \dot{x}_{\mathcal{H}} \quad \dot{v}_{\mathcal{R}} \quad \dot{v}_{\mathcal{H}}] = [v_{\mathcal{R}} \quad w_{\mathcal{R}} \quad v_{\mathcal{H}} \quad a_{\mathcal{R}} \quad a_{\mathcal{H}}], \quad (10)$$

where x (y) is the longitudinal (lateral) position, v (w) is the longitudinal (lateral) speed, and a is the acceleration. The sampling period is $\Delta t = 0.5[s]$. We use a state grid of the size $40 \times 6 \times 40 \times 6 \times 6$ to represent the discrete states of the human-robot system following [8]. The safety set $\mathbb{O}_{\text{safe}}$ includes states in which the boundaries of the two agents do not overlap. We use Δ=0.05 and $\Delta_\tau=\frac{1}{160}$ as the chance constraint thresholds in (9c). We let the highest intelligence level of the human be $k_{\max}$=2 based on experimental results in [44], [45]. The rationality coefficients take value from the set $\Lambda=\{0.5, 0.8, 1.0\}$. The planning horizon of the autonomous car is T=8.

Hierarchical planning and control. Behavioral planning and control of the autonomous car are hierarchically connected. The planning layer (Alg. 2) uses a low-fidelity model (10) to generate behavioral commands, and runs at $8Hz$ on a laptop with 2.8 GHz CPU. In the low-level control layer (running at $8Hz$), the vehicle dynamics are represented by a high-fidelity bicycle model [46], and we use a model predictive controller [47], [48] to generate continuous controls that drive the system to the desired states generated by Alg. 2. In Alg. 2, when the actual system state deviates from the state grid, the nearest neighboring grid will be exploited.

Baselines. We consider two baseline planners: 1) our planner without the feature of active information gathering, i.e., the autonomous car passively infers the human's latent states (BLP-1); 2) the strategic game-theoretic planner in [8] that treats the human-driven car as a *follower* who accommodates the actions from the autonomous car (BLP-2). Both BLP-2 and our planner use a closed-loop feedback structure when building human behavioral models, but our planner reasons about the heterogeneity in the human's cognitive limitations and irrationality through active inference rather than treating the human as a uniform follower.

B. The Human Behavioral Model

Human intelligence level interpretation. The ql-k model is exploited to reason about human behaviors under bounded intelligence. Recall that the level-0 agent represents a non-strategic agent who treats others as static obstacles. Thus, a ql-1 agent can be interpreted as a cautious agent since it believes that its opponent is an aggressive non-strategic agent. On the contrary, a ql-2 agent behaves aggressively since it believes its opponent is a cautious ql-1 agent.

Fig. 2 shows the interactions between two cars modeled as ql-k agents in the forced merging task. The heat-map displays the ql-k value function ($V^{*,i,k}$ described in Sec. IV-B) of the lower-lane car, indicating the preferred states; colder color means higher value. It can be observed in (a-b) that the interactions are seamless between a ql-1 agent and a ql-2 agent, which is expected since the ql-2 agent's belief in the model of the ql-1 agent matches the ground truth (note that the high-value region in the upper lane encourages the red car to merge ahead of the white car (a); the low-value region in front of the white car guides it to yield (b)). However, when an agent's true model deviates from the other agent's belief, conflicts may occur. For instance, (c) shows the interaction between two ql-1 agents with a dead-lock because both agents prefer to yield. The low-value region in front of the yellow car discourages it to merge although it is safe. Similarly, when two ql-2 agents interact (d), collisions may occur as both agents think their opponents will be likely to yield. Note that the high value region in the upper lane encourages the yellow car to merge even if it is not safe.

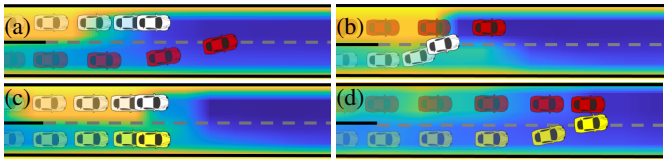

Fig. 2. **Interactions between ql-k agents**. (a-b): ql-1 (white) v.s. ql-2 (red); (c): ql-1 v.s. ql-1; (d): ql-2 v.s. ql-2. All agent have $\lambda = 1$, the same initial longitudinal position, and the same initial speed $12[m/s]$.

C. Case Studies and Quantitative Results

We compare our planner against the baselines via simulations. The human driven-car is modeled as a ql-k agent, and is also controlled by the hierarchical planning and control scheme described in Sec. VI-A, with behavioral commands obtained directly from the corresponding ql-k policies.

Hypotheses. We state the following two hypotheses: 1). active exploration improves efficiency of the robot's planning; 2). our planner is robust to the human's heterogeneous intelligence levels and irrational behaviors.

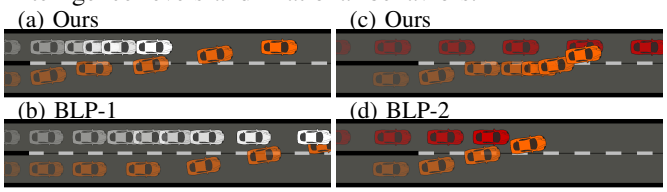

Fig. 3. **Planner comparison.** (a-b) show the interactions between a simulated *cautious* human-driven car (white) and the autonomous car (orange); (c-d) show those between a simulated *aggressive* human-driven car (red) and the autonomous car (orange). Initial speeds are 12 $[m/s]$.

We first show two case studies, highlighting the benefits that naturally emerged from our planner.

Scenario 1. Fig. 3(a-b) show a scenario where the autonomous car and a simulated cautious human-driven car (ql-1 agent) with rationality coefficient $\lambda = 0.8$ start at the same initial speed and longitudinal position. It can be observed that with our planner, the autonomous car actively indicates an intention to merge by nudging into the target lane, as it predicts that, the human's *reactions* triggered by the "probing" action can help disambiguate the human's latent states. With the baseline planner (BLP-1), the autonomous car takes a longer time to merge, since passive inference requires additional observations to infer the latent states.

Scenario 2. Fig. 3(c-d) show a case that is similar to scenario 1 but the human-driven car is simulated by an

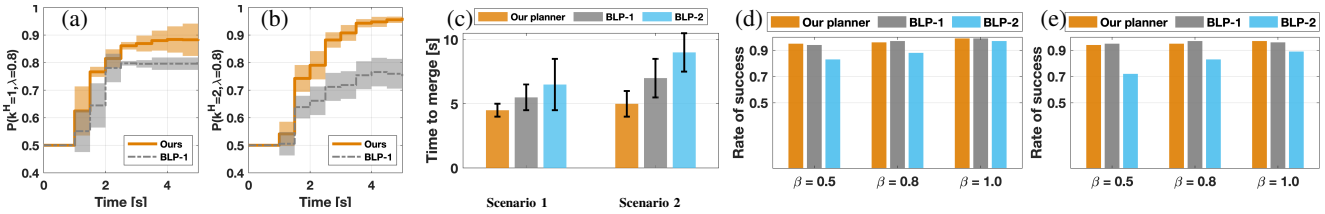

Fig. 4. **Evaluation result.** (a-b): Human latent states inference results for (a) cautious and (b) aggressive humans. (c): Average TM of the autonomous car; error bars show 95% confidence integral. (d-e): The rates of success under various human rationality coefficients (higher λ yields more rational human behaviors). The human-driven car is a simulated cautious driver with intelligence level-1 in (d), and an aggressive driver with intelligence level-2 in (e).

aggressive ql-2 agent who starts behind the autonomous car. With our planner, the autonomous car nudges in to explore the human's latent states, then quickly decides to yield after observing the humans' aggressive reactions. With BLP-2, the autonomous car initiates a dangerous merge as BLP-2 incorrectly assumes the human-driven car will likely yield. Note that when humans behave under heterogeneous cognitive states, it is critical for a robot to reason about such a heterogeneity to better predict human behaviors.

Metrics for quantitative results. In quantitative studies, we use two metrics for validating the hypotheses : 1) the rate of success (RS), which measures the percentage of simulations in which no collision or dead-lock occurs; 2) the time used by the autonomous car to complete the merge (TM).

Results. We run 50 simulations for each of the scenarios in the case studies using our planner and two baseline planners. We first compare our planner against BLP-1 in terms of inference performance (BLP-2 has no inference capability). In Fig. 4 (a-b), we show the time history of the autonomous car's belief in the ground truth human-driven car's latent states. It can be observed that our planner can more effectively identify the human's latent states by exploiting the mutual influence to make its human partner reveal his/her hidden states. In Fig. 4 (c), we show the average TM for the two scenarios. Note that our planner achieves the lowest TM, due to the active inference. Furthermore, our planner achieves the highest confidence on TM since our planner strategically generates actions that aim for triggering the most informative reactions from the human, this regulates human behaviors in a game-theoretic sense (such a phenomena is also observed in user studies Fig. 5(b)).

Next, we evaluate the robustness of our planner under various human intelligence levels and degrees of irrationality. We let the simulated ql-k human take his/her rationality coefficient from Λ and run 100 simulations for each (λ, k) combination. The human starts at a random position within 10 m around the autonomous car. In Fig. 4(d-e), we show the RS of each planner for each scenario. Note that in all cases, our planner and BLP-1 achieve more than 95% RS. This is attributed to reasoning about the human's bounded intelligence and irrationality, and enforcing the safety chance constraint. BLP-2 shows satisfactory RS when the simulated human is a ql-1 agent, but the RS decreases as the human becomes more irrational. When the human is a ql-2 agent, BLP-2 performs noticeably worse. The results indicate that our planner enables the robot to learn the human's latent states, plan more effectively, and be robust to the human's various intelligence levels and irrational behaviors.

D. User Study

Objective. We conduct user studies in which we let the autonomous car interact with a real human driver in a simulator (Fig. 1), showcasing the effectiveness of our planner.

Experiment setup. We recruited 10 human participants. For each participant, we ran Scenario 1 for 3 times using each of the planners. Here, the accelerations of the upper lane car are provided by human participants directly through a pedal.

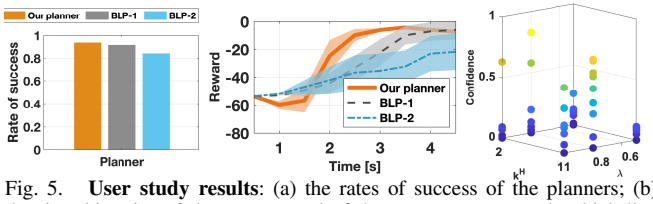

Fig. 5. **User study results**: (a) the rates of success of the planners; (b) the time histories of the step reward of the autonomous car; the thick line represents the mean and the shaded area represents the 95% confidence tube of the data; (c) the distribution of inferred intelligence levels and rationality coefficients of the human participants.

Results. Fig. 5(a) shows the RS of each planner. It can be observed that both our planner and BLP-1 outperform the BLP-2 in terms of safety. Fig. 5(b) shows the time histories of the step reward collected by the autonomous car. Note that although the autonomous car pays more costs in the first few steps using our planner (due to the probing actions) it is able to complete the task much more effectively and safely compared with the baselines. In Fig. 5(c), we show the distributions of inferred intelligence levels and rationality coefficients. It can be observed that roughly 60% of the participants are identified as ql-1 agents, which is aligned with the experiment study on other forms of games [45]. In addition, Fig. 5(c) suggests that, under the reward function assumed by the autonomous car, most of the participants demonstrated behaviors that align with the behaviors produced by the rationality coefficient $\lambda = 0.8$.

VII. Conclusion

We proposed an anytime game-theoretic planning framework that integrates iterative human reasoning models, a POMDP, and Monte-Carlo belief tree search for robot behavioral planning. Our approach equips a robot with the ability to reason about its human partner's latent cognitive states (bounded intelligence and irrationality) and enables the robot to safely and actively learn these latent states to better maximize its utility. We applied the proposed approach to an autonomous driving domain where our behavioral planner and a low-level motion controller hierarchically control an autonomous car to negotiate traffic merges. Both simulation and user study results demonstrated the effectiveness of our planner compared with baseline planners.

REFERENCES

[1] R. B. Myerson, *Game theory*. Harvard university press, 2013.

[2] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[3] J. K. Goeree and C. A. Holt, "Ten little treasures of game theory and ten intuitive contradictions," *American Economic Review*, vol. 91, no. 5, pp. 1402–1422, 2001.

[4] V. P. Crawford and N. Iriberri, "Level-k auctions: Can a nonequilibrium model of strategic thinking explain the winner's curse and overbidding in private-value auctions?" *Econometrica*, vol. 75, no. 6, pp. 1721–1770, 2007.

[5] J. R. Wright and K. Leyton-Brown, "Level-0 meta-models for predicting human behavior in games," in *Proceedings of the fifteenth ACM conference on Economics and computation*, 2014, pp. 857–874.

[6] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 66–73.

[7] 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.

[8] 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.

[9] E. Stefansson, J. F. Fisac, D. Sadigh, S. S. Sastry, and K. H. Johansson, "Human-robot interaction for truck platooning using hierarchical dynamic games," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3165–3172.

[10] N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 3215–3220.

[11] 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.

[12] S. Li, N. Li, A. Girard, and I. Kolmanovsky, "Decision making in dynamic and interactive environments based on cognitive hierarchy theory, bayesian inference, and predictive control," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 2181–2187.

[13] R. Tian, N. Li, I. Kolmanovsky, and A. Girard, "Beating humans in a penny-matching game by leveraging cognitive hierarchy theory and bayesian learning," in *2020 American Control Conference (ACC)*, 2020, pp. 4652–4657.

[14] M. Kwon, M. Li, A. Bucquet, and D. Sadigh, "Influencing leading and following in human-robot teams." in *Robotics: Science and Systems*, 2019.

[15] M. Simaan and J. B. Cruz, "Additional aspects of the stackelberg strategy in nonzero-sum games," *Journal of Optimization Theory and Applications*, vol. 11, no. 6, pp. 613–626, 1973.

[16] M. Costa-Gomes, V. P. Crawford, and B. Broseta, "Cognition and behavior in normal-form games: An experimental study," *Econometrica*, vol. 69, no. 5, pp. 1193–1235, 2001.

[17] J. Pineau, G. Gordon, S. Thrun, *et al.*, "Point-based value iteration: An anytime algorithm for pomdps," in *IJCAI*, vol. 3, 2003, pp. 1025–1032.

[18] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-based value iteration for continuous pomdps," *Journal of Machine Learning Research*, vol. 7, no. Nov, pp. 2329–2367, 2006.

[19] H. Kurniawati, D. Hsu, and W. S. Lee, "Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces." in *Robotics: Science and systems*, vol. 2008. Zurich, Switzerland., 2008.

[20] C. Yu, J. Chuang, B. Gerkey, G. Gordon, and A. Ng, "Open-loop plans in multi-robot pomdps," Technical report, Stanford CS Dept, Tech. Rep., 2005.

[21] A. Weinstein and M. L. Littman, "Open-loop planning in large-scale stochastic domains," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013, pp. 1436–1442.

[22] D. Perez Liebana, J. Dieskau, M. Hunermund, S. Mostaghim, and S. Lucas, "Open loop search for general video game playing," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015, pp. 337–344.

[23] T. Phan, L. Belzner, M. Kiermeier, M. Friedrich, K. Schmid, and C. Linnhoff-Popien, "Memory bounded open-loop planning in large pomdps using thompson sampling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7941–7948.

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

[25] M. H. Lim, C. J. Tomlin, and Z. N. Sunberg, "Sparse tree search optimality guarantees in pomdps with continuous observation spaces," *arXiv preprint arXiv:1910.04332*, 2019.

[26] C. F. Camerer, T.-H. Ho, and J.-K. Chong, "A cognitive hierarchy model of games," *The Quarterly Journal of Economics*, vol. 119, no. 3, pp. 861–898, 2004.

[27] D. O. Stahl II and P. W. Wilson, "Experimental evidence on players' models of other players," *Journal of economic behavior & organization*, vol. 25, no. 3, pp. 309–327, 1994.

[28] J. R. Wright and K. Leyton-Brown, "Predicting human behavior in unrepeated, simultaneous-move games," *Games and Economic Behavior*, vol. 106, pp. 16 – 37, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0899825617301574

[29] R. D. McKelvey and T. R. Palfrey, "Quantal response equilibria for normal form games," *Games and economic behavior*, vol. 10, no. 1, pp. 6–38, 1995.

[30] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, "A real-time game theoretic planner for autonomous two-player drone racing," *Robotics: Science and Systems*, 2018.

[31] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.

[32] A. Feldbaum, "Dual control theory. i," *Avtomatika i Telemekhanika*, vol. 21, no. 9, pp. 1240–1249, 1960.

[33] ——, "Dual control theory. ii," *Avtomatika i Telemekhanika*, vol. 21, no. 11, pp. 1453–1464, 1960.

[34] Y. Bar-Shalom and E. Tse, "Dual effect, certainty equivalence, and separation in stochastic control," *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 494–500, 1974.

[35] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[36] B. Wittenmark, "Adaptive dual control methods: An overview," in *Adaptive Systems in Control and Signal Processing 1995*, ser. IFAC Postprint Volume, C. BÁNYÁSZ, Ed. Oxford: Pergamon, 1995, pp. 67 – 72. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978008042375350010X

[37] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[38] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5876.

[39] M. Ono, Y. Kuwata, and J. Balaram, "Joint chance-constrained dynamic programming," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 1915–1922.

[40] D. Isele, "Interactive decision making for autonomous vehicles in dense traffic," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3981–3986.

[41] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[42] 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.

[43] R. Tian, L. Sun, and M. Tomizuka, "Bounded risk-sensitive markov game and its inverse reward learning problem," *arXiv preprint arXiv:2009.01495*, 2020.

[44] M. A. Costa-Gomes and V. P. Crawford, "Cognition and behavior in two-person guessing games: An experimental study," *American Economic Review*, vol. 96, no. 5, pp. 1737–1768, Dec. 2006.

[45] M. A. Costa-Gomes, V. P. Crawford, and N. Iriberri, "Comparing models of strategic thinking in van huyck, battalio, and beil's coordination games," *Journal of the European Economic Association*, vol. 7, no. 2-3, pp. 365–376, 2009.

[46] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1094–1099.

[47] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[48] J. Mattingley, Y. Wang, and S. Boyd, "Code generation for receding horizon control," in *2010 IEEE International Symposium on Computer-Aided Control System Design*, 2010, pp. 985–992.