Multi-objective optimal control for proactive decision-making with temporal logic models

Sandeep P. Chinchali, Scott C. Livingston, and Marco Pavone

Abstract The operation of today's robots increasingly entails interactions with humans, in settings ranging from autonomous driving amidst human-driven vehicles to collaborative manufacturing. To effectively do so, robots must proactively decode the intent or plan of humans and concurrently leverage such a knowledge for safe, cooperative task satisfaction—a problem we refer to as proactive decision making. However, the problem of proactive intent decoding coupled with robotic control is computationally intractable as a robot must reason over several possible human behavioral models and resulting high-dimensional state trajectories. In this paper, we address the proactive decision making problem using a novel combination of algorithmic and data mining techniques. First, we distill high-dimensional state trajectories of human-robot interaction into concise, symbolic behavioral summaries that can be learned from data. Second, we leverage formal methods to model high-level agent goals, safe interaction, and information-seeking behavior with temporal logic formulae. Finally, we design a novel decision-making scheme that simply maintains a belief distribution over high-level, symbolic models of human behavior, and proactively plans informative control actions. Leveraging a rich dataset of real human driving data in crowded merging scenarios, we generate temporal logic models and use them to synthesize control strategies using tree-based value iteration and reinforcement learning (RL). Results from two simulated self-driving car scenarios, one cooperative and the other adversarial, demonstrate that our data-driven control strategies enable safe interaction, correct model identification, and significant di-

Sandeep P. Chinchali
Stanford University, Stanford, CA, USA, e-mail: csandeep@stanford.edu

Scott C. Livingston
e-mail: slivingston@cds.caltech.edu

Marco Pavone
Stanford University, Stanford, CA, USA e-mail: pavone@stanford.edu

The authors were partially supported by the Office of Naval Research, ONR YIP Program, under Contract N00014-17-1-2433.

mensionality reduction.

Keywords: Decision-making, formal methods, human-robot interaction, data-mining

1 Introduction

Data-driven learning and proactive decision making are key ingredients of modern autonomous systems (AS). Robots, ranging from surgical assistants to autonomous cars, must seamlessly interact with other agents, which requires understanding their intents and behavioral models. While most current strategies used by a robot to understand the plan of a human rely on passive observations, recent work has focused significant attention on *proactive* intent decoding and decision making [16, 9]. Examples include autonomous cars that gently nudge into adjacent lanes to discern the driving style of nearby drivers for lane-merging [16] or use large signs to proactively signal when pedestrians can safely cross at intersections [9].

A principal challenge of *proactive* decision making coupled with *concurrent* robotic control is that the resulting decision-making problem is computationally intractable. A robot must optimize over several plausible models of human behavior, which is especially complex if we consider a high-dimensional set of trajectories that an agent may enact to accomplish its goals. In this context, the principal aim of this paper is to provide a tractable approach for proactive decision making that exploits algorithmic and data mining techniques for dimensionality reduction.

Related work: Prior work has typically *separately* treated the problems of intent decoding and strategy synthesis, which describes how to best use learned information for planning future actions. In [17], this gap is partially bridged by modeling interdependency of human-robot planning using Gaussian processes. Recently, Sadigh et al. [16] show how a robotic car can identify whether nearby human drivers are aggressive or cautious by nudging into adjacent lanes for information gain. Though promising, the scheme does not account for safety constraints in probe selection and assumes a static human driving style. A key motivation of our work is to incorporate safety constraints and anticipate a rich variety of human behaviors that may contextually change based on robot interaction to enhance autonomy.

Proactive decision making can be cast as a Partially Observable Markov Decision Process (POMDP) where the hidden mode of a human must be estimated by a robot, but POMDPs can only be solved efficiently for small problems [13, 11]. Relevant prior work using POMDPs includes hindsight optimization for grasping [7], interactive POMDPs (I-POMDPs) [6], and goal decomposition approaches [12]. Recent work on temporal logic models highlights their ability to capture safety constraints and high-level interactions [18, 8, 14]. In this paper, we use temporal logic as a tool for dimensionality reduction by distilling complex human-robot interactions into succint behavioral templates, which we learn from real driving data.

Statement of contributions: We significantly reduce the computational complexity of proactive decision making using a novel combination of formal methods and data mining. First, we show how to filter high-dimensional state trajectories into a concise set of behavioral models expressed using temporal logic formulae. We construct concise states as belief distributions over formal, symbolic models, as opposed to beliefs over a much higher-dimensional set of state trajectories. Leveraging real human driving data from the Stanford Drone Dataset (SDD) [15], we mine parameters for temporal logic formulae that we select to be representative of key

lane-merging behavior. Our framework, however, is general and can be extended to formulae that are automatically learned from data. Based on these symbolic models, we synthesize value iteration and reinforcement learning (RL) controllers that proactively probe human intent for information gain while minimizing control cost. Simulated studies of two robotic car scenarios, ranging from adversarial to cooperative, validate our approach, which we also characterize theoretically.

Paper organization: The rest of the paper is organized as follows. In Section 2, we introduce two motivating examples of proactive decision making. We then introduce temporal logic and our solution framework in Section 3. Next, we show how to reduce problem complexity and provide a theoretical analysis of our solution framework in Section 4. Sections 5 and 6 provide simulations from data-driven models from the SDD and control strategies generated by both value iteration and reinforcement learning. Finally, we provide concluding remarks in Section 7.

2 Examples of Proactive Decision Making

Throughout this paper, we refer to the following two examples of proactive decision making to illustrate key technical concepts.

Example 1, Cooperative lane-merging: A robotic car must merge into a crowded roundabout with pedestrians and bikers, such as in Figure 1(a) from the SDD [15]. Inspired by autonomous car startup *drive.AI*'s [9] recent proposal, the robotic car can *proactively* instruct pedestrians to wait, safely cross an intersection, or choose not to signal. Pedestrians obey or disobey the robot's safety indication and cross based on their observations of traffic and internal risk profile (cautious or daring). The robot balances the cost of signaling, which represents a risk probability of erroneously indicating safe conditions, with exploitation of its current pedestrian model. Notably, we mine key temporal logic formulae for this scenario from the SDD.

Example 2, Adversarial car-pursuit: In Figure 1(b), a robotic aid vehicle (green) is transporting medical supplies in an urban warzone, where it might be followed by benign civilian vehicles, an enemy surveillance car, or be directly chased by an enemy pursuer (red). If the robot *proactively* makes subtle route changes, it can differentiate benign civilian cars from surveillance vehicles since it is highly improbable civilians systematically follow the robot. Thus, the robot must balance exploration of follower intent, which comes with a control cost of extra travel time and fuel, with exploitation of its currently assumed model for safe delivery of supplies.

3 Proactive Decision Making Framework

In this section, we formulate the problem of proactive decision making with formal methods. First, a definition is presented for sequences of interaction between multiple agents. Next, the specification language used throughout the paper is introduced, followed by an *adversarial* Markov Decision Process (MDP) with labelings that allow it to be evaluated with respect to a specification. Finally, a problem is formulated for strategy selection and optimality of the adversarial MDP.

Consider a set of m agents operating in discrete time. Let $\mathscr{A} = \{1, \ldots, m\}$ denote the set of agents, $\text{Act} = \text{Act}_1 \times \cdots \times \text{Act}_m$ denote their joint action space, and $\mathbf{S}$ denote a joint state space.

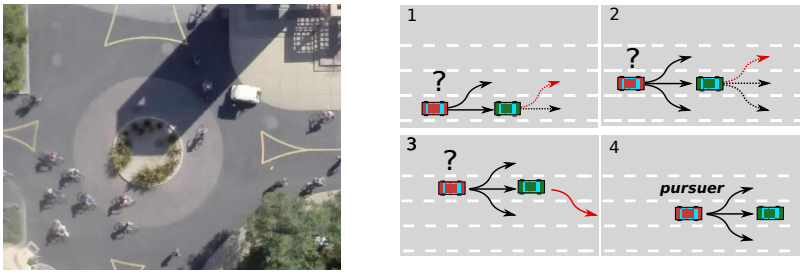

Fig. 1: Examples of Proactive Decision Making: (Left) A cooperative scenario, from the Stanford Drone Dataset [15], shows how cars must nudge into crowded roundabouts. (Right) An adversarial scenario, inspired by [5], shows how a green robotic car must safely swerve lanes to determine if it is being pursued by a red adversarial car.

Definition 1. An *interaction sequence* is a sequence of state-action pairs indexed by time and denoted by

$$\mathscr{H}_t(\tau) = \left[(s^\tau, \mathbf{a}^\tau), \ldots, (s^{\tau+t-1}, \mathbf{a}^{\tau+t-1}), s^{\tau+t}\right],$$

where $\mathscr{H}_t(\tau)$ is said to begin at discrete time τ and have duration t, and where states are given by $s \in \mathbf{S}$ and actions by $\mathbf{a} = (a_1, \ldots, a_m) \in \text{Act}$. An interaction sequence $\mathscr{H}_t(0)$ is called an *interaction history* and is also written $\mathscr{H}_t$.

In the car-following example, the set of agents comprises the robot and follower, states are the lane occupancies, and actions are probing route deviations.

3.1 Bounded linear-time temporal logic (BLTL)

In this paper, formal specifications of interaction sequences are defined for *finite* durations that are well-suited to information gathering tasks. We employ a fragment of metric temporal logic (MTL) named bounded linear-time temporal logic (BLTL),which was introduced in [5] and summarized below. The crux of BLTL is to first define the operator $\mathbf{U}_I$, where $I = [a,b]$ is a bounded interval on the nonnegative integers $\mathbb{N}$, to express constrained reachability over finite durations of non-dense time. For Boolean formulae f and g that do not contain temporal operators, $f\,\mathbf{U}_{[a,b]}\,g$ is satisfied by a sequence σ if f is true at each state beginning at time a, until a state is reached where g is satisfied or the time b is reached. A key feature of all BLTL formulae is that they can be decided using a finite sequence of timesteps, since all such intervals I are bounded.

To reason about interaction sequences, we need a mechanism to check if high-level logical formulae, constructed from a set of *atomic propositions*, are satisfied at individual states. We denote a finite set of *atomic propositions* by Π, where elements of Π are Boolean-valued variables that, at each discrete time, evaluate to either `True` or `False`. Atomic propositions associated with a self-driving car might include $\mathscr{Y} = \{C_x\}_{x=1}^{X}$ to indicate the robot occupies lane x of X total lanes.

BLTL syntax over interval I is given by the context-free grammar

$$\varphi ::= \texttt{True} \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \varphi\,\mathbf{U}_I\,\varphi, \tag{1}$$

where p is an atomic proposition $p \in \Pi$. Here, atomic propositions p can be combined to describe logical formulae φ by using standard logical connectives such as conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), and implication ($\Longrightarrow$), coupled with *temporal* operators such as eventually ($\Diamond_I$), always ($\Box_I$), and until ($\mathbf{U}_I$). The connective $\Box_I \varphi$ means that φ is true at all positions of the word in the interval I of time steps; the connective $\Diamond_I \varphi$ means that φ eventually becomes true within a finite time; the connective $\varphi_1 \mathbf{U} \varphi_2$ means that φ_1 has to hold at each position in the word, at least until φ_2 is true in interval I. Significant expressivity can be achieved by combining temporal and Boolean operators in BLTL.

For a search and rescue mission triggered by a flare, the operational behavior "once a flare is lighted, always a drone is dispatched until a human-operated rescue helicopter arrives within interval T_h" can be expressed as:

$$\texttt{flare} \Longrightarrow (\texttt{drone}\ \mathbf{U}_{[0,T_h]}\ \texttt{helicopter}).$$

Interaction sequences (words) must be long enough to decide whether a BLTL formula is satisfied. For any BLTL formula φ, $T(\varphi)$ denotes the minimum time at which the satisfaction of φ by any interaction sequence can be decided, such as T_h in the drone example. $T(\varphi)$ is constant, finite, always exists for each φ, and is no longer than the sum of the upper bounds of all intervals appearing in φ. More details on the semantics of BLTL are in [5] and on topics in model checking in [2].

3.2 High-level agent intent models

Henceforth we treat the case of two interacting agents, one of which we control. The other agent is referred to as the *adversary*. To reason about their interaction, we use *labelled adversarial Markov Decision Processes (aMDPs)*, which are defined similarly as in [18].

Definition 2. : A *labelled adversarial MDP* $\mathscr{M}$ is a tuple $(\mathbf{S}, \mathrm{Init}, \mathrm{Act}^c, \mathrm{Act}^u, \mathbf{P}, \Pi, L)$, where $\mathbf{S}$ is a finite set of states, $\mathrm{Init} \subseteq \mathbf{S}$ is a set of possible initial states, Act^c is a mapping from states into sets of *controlled actions*, Act^u is a mapping from states into sets of *uncontrolled actions* (or *adversarial actions*), Π is a finite set of atomic propositions, the labelling function $L : \mathbf{S} \to 2^{\Pi}$ maps states to atomic propositions, and $\mathbf{P} : \mathbf{S} \times \mathrm{Act}^c \times \mathrm{Act}^u \times \mathbf{S} \to [0,1]$ defines transition probabilities where for each state $s \in \mathbf{S}$, $a \in \mathrm{Act}^c(s)$, and $b \in \mathrm{Act}^u(s)$, $\sum_{s' \in \mathbf{S}} \mathbf{P}(s,a,b,s') = 1$.

Assumption 1 *For every state $s \in \mathbf{S}$, $\mathrm{Act}^c(s) \neq \emptyset$ and $\mathrm{Act}^u(s) \neq \emptyset$.*

Intuitively, this assumption stipulates that no dead-ends exist.

3.3 Interaction sequences and traces of adversarial MDPs

Let $\mathscr{M}$ be a labelled adversarial MDP. A strategy is a partial function from $\mathrm{Hist}(\mathscr{M}, T)$ to exactly one of the two action sets associated with $\mathscr{M}$: Act^c, Act^u. Let π be a strategy mapping into Act^c, and let μ be a strategy mapping into Act^u. The set of interaction histories consistent with these strategies is defined by

$$\mathrm{Hist}(\mathscr{M}, T, \pi, \mu) = \big\{ \mathscr{H}_T \mid s^0 \in \mathrm{Init} \quad \wedge \\ \forall \tau < T : \mathbf{P}(s^\tau, \pi(\mathscr{H}_\tau), \mu(\mathscr{H}_\tau), s^{\tau+1}) > 0 \big\}. \quad (2)$$

Using the labelling associated with $\mathscr{M}$, the set of traces that may occur under strategies π and μ is defined by

$$\text{Traces}(\mathscr{M},T,\pi,\mu) = \big\{\sigma \in \Sigma^{T+1} \mid \exists \mathscr{H}_T \in \text{Hist}(\mathscr{M},T,\pi,\mu) : \forall \tau : 0 \leq \tau \leq T \wedge \sigma_\tau = L(s^\tau)\big\}, \quad (3)$$

where $\Sigma = 2^{\Pi}$. In words, Σ is the set of subsets of atomic propositions. An atomic proposition q is said to be true at a state s if and only if $q \in L(s)$. The dependence of each $\sigma \in \text{Traces}(\mathscr{M},T,\pi,\mu)$ in (3) on some $\mathscr{H}_T \in \text{Hist}(\mathscr{M},T,\pi,\mu)$ can be generalized to show there is a function $\mathscr{L}$ from $\text{Hist}(\mathscr{M},T,\pi,\mu)$ onto $\text{Traces}(\mathscr{M},T,\pi,\mu)$ consistent with the comprehension in (3), i.e., such that for all $\mathscr{H}_T \in \text{Hist}(\mathscr{M},T,\pi,\mu)$, $\mathscr{L}(\mathscr{H}_T) \in \text{Traces}(\mathscr{M},T,\pi,\mu)$ and $\mathscr{H}_T$ realizes the existential quantification in (3).

3.4 Probability of satisfaction and agent-intent models

Let φ be a BLTL formula φ defined in terms of atomic propositions from Π. Recall that there is a minimal bound $T(\varphi)$ such that for any $T \geq T(\varphi)$ and for any $\sigma \in \Sigma^T$, it is decided whether $\sigma \models \varphi$ or $\sigma \not\models \varphi$, i.e., the word σ has sufficiently many positions to decide satisfaction of φ. Now, define the expression $\mathscr{H}_T \models \varphi$ to be true if and only if $\mathscr{L}(\mathscr{H}_T) \models \varphi$, which indeed is well-defined for $T \geq T(\varphi)$ because $\text{Traces}(\mathscr{M},T,\pi,\mu) \subseteq \Sigma^{T+1}$.

Recall that the aMDP $\mathscr{M}$ has finite state and action sets. For $T < \infty$, it follows that the set $\text{Hist}(\mathscr{M},T,\pi,\mu)$ is finite for any policies π and μ. Entirely similar observations hold for $\text{Hist}(\mathscr{M},T)$ and $\text{Traces}(\mathscr{M},T,\pi,\mu)$.

Recall that each interaction history $\mathscr{H}_T \in \text{Hist}(\mathscr{M},T,\pi,\mu)$ defines a finite sequence of states and actions. From (2), it follows that $\mathscr{H}_T$ is equivalent to a finite sequence of state-action-state triples, each of which is a transition of $\mathscr{M}$. Recall that the probability of each transition is defined by $\mathbf{P}$. Computing the product of these probabilities for each element of $\text{Hist}(\mathscr{M},T,\pi,\mu)$ and normalizing provides a probability mass function over $\text{Hist}(\mathscr{M},T,\pi,\mu)$.

As shown earlier, for sufficiently large T, it can be decided for each element of $\text{Hist}(\mathscr{M},T,\pi,\mu)$ whether it satisfies the BLTL formula φ, hence we define

$$P_{\mathscr{M},\pi,\mu}(\text{Init} \models \varphi) \quad (4)$$

as the probability of satisfying φ when AS strategies and adversarial strategies are applied to $\mathscr{M}$. Notice that T does not need to be explicitly given above because once satisfaction of φ is decided for an interaction history, any additional actions cannot change the outcome. When Init is a singleton, we write $P_{\mathscr{M},\pi,\mu}(s \models \varphi)$.

We now define high-level agent intent models and assumptions on their behavior.

Definition 3. An *agent-intent model* is a pair $(\mathscr{M},\varphi)$, where $\mathscr{M}$ is a labelled adversarial MDP and φ is a BLTL formula.

The following assumption states that the adversary, if following model $(\mathscr{M},\varphi)$, will select a strategy that maximizes the probability of satisfying φ. Such an assumption is natural since the adversary's own strategy must be as concordant as possible with its true specification, but we allow for measurement uncertainty etc.

Assumption 2 *Let $(\mathscr{M},\varphi)$ be an agent-intent model. For any $s \in$ Init, the adversarial strategy is from the set*

$$\arg\max_{\mu}\min_{\pi} P_{\mathscr{M},\pi,\mu}(s \models \varphi),$$

where $P_{\mathscr{M},\pi,\mu}(s \models \varphi)$ is the probability that the labeled Markov chain induced by finite-memory strategies π and μ and with initial state s satisfies φ.

3.5 Proactive Decision Making with BLTL constraints

We now introduce the problem of Proactive Decision Making with BLTL formulae and adversarial MDPs. It involves interaction between an autonomous system (AS) and an adversary, referred to as `adv`. The AS must find a control policy $\pi_{\texttt{AS}}$ that maximizes reward R despite possible interference from the adversary enacting controller $\pi_{\texttt{adv}}$. Reward function $R : \mathscr{H}_t \mapsto \mathbb{R}$ defines a scalar reward over interaction history $\mathscr{H}_t$ which incentivizes the AS to learn about the true model M_j of agent `adv` while minimizing the cost of information gain. Interaction histories must satisfy goal and safety specifications encoded in BLTL formula φ.

Problem 1. ProDM-BLTL: Proactive Decision Making with BLTL:
Given a finite set of agent-intent models $\{(\mathscr{M}_1,\varphi_1),\ldots,(\mathscr{M}_N,\varphi_N)\}$, find strategy $\pi^*_{\texttt{AS}}$ such that

$$\pi^*_{\texttt{AS}} \in \arg\max_{\pi_{\texttt{AS}}}\left(\min_{\pi_{\texttt{adv}}} \mathbb{E}_{\mathscr{M}_j,\pi_{\texttt{AS}},\pi_{\texttt{adv}}}\left(R(\mathscr{H}_T)\right)\right)$$

where ground-truth model j is fixed and known to the agent `adv` but is unknown to the AS, and where $\pi_{\texttt{adv}} \in \arg\max_{\mu} P_{\mathscr{M}_j,\pi_{\texttt{AS}},\mu}(\text{Init} \models \varphi_j)$.

In terms of the ongoing example of the car-following scenario, R is a weighted sum of fuel and lane deviation cost coupled with a metric of information gain and specifications φ govern how different follower car models pursue the robot. Interaction sequences $\mathscr{H}_T$ are joint robot-follower lane trajectories.

4 Dimensionality Reduction and Solution Algorithm

Problem 1 is intractable because, as a consequence of not knowing the ground-truth index j, we must optimize over all possible agent-intent models $M_i = (\mathscr{M}_i,\varphi_i)$ and interaction sequences $\mathscr{H}_t$. In this section, we present, solve, and theoretically analyze a new problem based on complexity reductions from Problem 1.

4.1 Request-response formulae and satisfaction bitvectors

For a special form of BLTL formulae that is defined below, we are able to reduce the size of the action space of the abstract system. Let $\{(\mathscr{M}_1,\varphi_1),\ldots,(\mathscr{M}_N,\varphi_N)\}$ be a set of agent-intent models (recall Definition 3 in Section 3.4).

Definition 4. (BLTL request-response formula): The specification φ_i for agent-intent model i is said to be a *BLTL request-response formula* if it has the form

$$\varphi_i = \bigwedge_{r\in\{1,\ldots,R_i\}}\left(\psi^{\text{req},r}_{AS,i} \implies \Diamond_{[0,T_r]}\,\psi^{\text{res},r}_i\right) = \bigwedge_{r\in\{1,\ldots,R_i\}}\varphi^r_i \tag{5}$$

where each mode r represents an AS (robot) *probe* to agent i given by $\psi_{AS,i}^{\text{req},r}$, $\psi_i^{\text{res},r}$ is the *informative response*, and T_r is the maximum duration to wait for the response.

BLTL request-response formulae are practically-motivated because many real-life information-gathering tasks generate a response within a bounded time. Autonomous cars must wait a finite duration for pedestrians or anomalous traffic patterns to reach normal states and deadlock situations rarely persist indefinitely. In the car-following example, a request is a lane-change and an informative response is for an adversary to follow the robot within a bounded interval T_r. In the definition of belief MDP given later in this section, request-response formulae allow us to define an abstract "impulse" action, thus reducing the effective space of possible actions for policy selection.

Given our goal of reasoning about likelihood among agent-intent models, the effective state space for planning can be reduced by abstracting it into outcomes of interaction, which we now define.

Definition 5. (Satisfaction bitvector observation): Consider N high-level agent intent models $M_i = (\mathscr{M}_i, \varphi_i)$ (Def. 3 in Section 3.4) where each specification φ_i has R_i BLTL request-response formulae. Define $Q = \sum_i R_i$, and $T = \max_i T(\varphi_i)$. The *satisfaction bitvector observation* o is a function that maps an interaction history $\mathscr{H}_T$ to a vector of dimension Q where the q-th element is 1 if $\mathscr{H}_T$ satisfies the qth interaction formula $\varphi_q = \varphi_i^r$ and is otherwise 0. Observation space $\mathscr{O} = \{o \mid o \in \{0,1\}^Q\}$ comprises 2^Q possible bitvectors. Written in terms of the indicator function $\mathbb{1}$,

$$o(\mathscr{H}_T) = [\mathbb{1}\,(\mathscr{H}_T \models \varphi_1^1), \ldots, \mathbb{1}\,(\mathscr{H}_T \models \varphi_N^{r_N})]. \tag{6}$$

4.2 Reduced ProDM-BLTL Problem with Belief MDPs

Leveraging the previous definitions, we reduce the complexity of Problem 1 by constructing a new reward function and belief MDP (defined below) where states are belief distributions over the N candidate agent-intent models, actions are BLTL requests, and transitions depend on satisfaction bitvector observations.

The *belief MDP* of a set of candidate agent-intent models is defined as $\tilde{M} = (\tilde{B}_0, \tilde{B}, \tilde{A}, P_b, \tau_b)$ with the following components:

1. $\tilde{B} = \{B \in \mathbb{R}^N \mid B(i) \geq 0 \wedge \sum_i B(i) = 1\}$.
   Belief states are probability distributions over the N candidate models.
2. $\tilde{A} = \{\psi_{AS,i}^{\text{req},r} \mid i \in \{1,\ldots,N\}, r \in \{1,\ldots,R_i\}\}$.
   Abstract (impulse) actions are controlled requests for any BLTL request-response formula.
3. $P_b : \tilde{B} \times \tilde{A} \times \tilde{B} \to \mathbb{R}$
   Belief transition function P_b is defined as follows, where $o_k = o(\mathscr{H}_T(t_k))$ is a satisfaction bitvector observation (Def. 5 in Section 4.1).

$$\begin{aligned} P_b(B_k, a_k, B_{k+1}) &= \Pr(B_{k+1} \mid B_k, a_k) \\ &= \sum_{o_k \in \mathscr{O}} \Pr(B_{k+1} \mid B_k, a_k, o_k) \sum_{i=1}^{N} \Pr(o_k \mid a_k, \mathscr{M}_i) B_k(i) \end{aligned}$$

where

$$\Pr(o_k \mid a_k, \mathscr{M}_i) = \prod_{q=1}^{Q} \Big(o_k^q \Pr(\mathscr{H}_T(t_k) \models \varphi_q \mid a_k, \mathscr{M}_i) + (1 - o_k^q) \Pr(\mathscr{H}_T(t_k) \not\models \varphi_q \mid a_k, \mathscr{M}_i) \Big), \quad (7)$$

where $o_k = (o_k^1, \ldots, o_k^Q)$ and φ_q was in Def. 5. Notice that $\Pr(o_k \mid a_k, \mathscr{M}_i)$ is the joint probability of satisfaction or not, depending on respective elements of the bitvector o_k, for each agent-intent model, given that the true model is $(\mathscr{M}_i, \varphi_i)$.

4. The belief update function $\tau_b : \tilde{B} \times \tilde{A} \times \mathscr{O} \to \tilde{B}$ maps belief state B_k, action a_k, and resulting observation o_k to new belief vector B_{k+1} by Bayes' Rule. The belief update depends on the probability of observation o_k given an informative impulse a_k under each competing model M_i, i.e., $B_{k+1} = \tau_b(B_k, a_k, o_k)$ where

$$B_{k+1}(i) = \eta B_k(i) \Pr(o_k \mid a_k, \mathscr{M}_i), \quad (8)$$

and η is a normalization factor.

5. The initial state is a uniform probability distribution over N possible models, i.e., $\tilde{B}_0 = B_0 = [\frac{1}{N}, \ldots, \frac{1}{N}]$.
6. The stage reward is a weighted sum of control cost and information gain, where information gain is measured as entropy reduction in the belief vector

$$R_k^H(B_k, a_k, B_{k+1}) = -\alpha c(a_k, B_k) + \beta [H(B_k) - H(B_{k+1})]. \quad (9)$$

Here, $c : \tilde{A} \times \tilde{B} \to \mathbb{R}$ is the control cost function, H is the Shannon entropy, and $\alpha > 0, \beta > 0$ weight the control cost and information gain.

A transition of $\tilde{M}$ occurs in two parts starting at an iteration k. First, we start at a belief state B_k at time t_k when the fully observable aMDP state is s_{t_k}. An action is selected as an impulse a_{t_k}, corresponding to a specific BLTL request-response formula for some model i and high-level mode r. By selecting this action, we can assume model i is the true model and synthesize a strategy corresponding to aMDP $\mathscr{M}_i$ and interaction formula φ_i, using Assumption 2 in Section 3.4.

Second, a finite-horizon play occurs, resulting in a final state $s_{t_k+T(\varphi)}$ of the aMDP and an interaction sequence from t_k of length $T(\varphi)$, $\mathscr{H}_T(t_k)$. The interaction sequence can then be mapped to a satisfaction bitvector observation $o_k = o(\mathscr{H}_T(t_k))$. Given current belief state B_k, action a_k, and satisfaction bitvector observation o_k, we can construct a new belief state B_{k+1} by Bayes' Rule. From new belief state B_{k+1} and underyling aMDP state $s_{t_k+T(\varphi)}$, the process repeats.

We can now present the ProDM-BLTL problem solved in this paper.

Problem 2. Belief-ProDM: ProDM-BLTL with Belief MDPs
Let $\{(\mathscr{M}_1, \varphi_1), \ldots, (\mathscr{M}_N, \varphi_N)\}$ be a set of agent-intent models, and let $(\tilde{B}_0, \tilde{B}, \tilde{A}, P_b, \tau_b)$ be the belief MDP corresponding to it. Given the reward function R_k^H in (9) and discounting factor $\gamma < 1$, solve

$$\begin{aligned} \max_{\xi} \quad & \lim_{K\to\infty} \mathbb{E}\left(\sum_{k=0}^{K} \gamma^k R_k^H(B_k, a_k, B_{k+1})\right) \\ \text{such that} \quad & o_k = o(\mathscr{H}_T(t_k)) \\ & B_{k+1} = \tau_b(B_k, a_k, o_k) \\ & \text{for all } k = 0, \ldots, K-1 \end{aligned}$$

4.3 Value Iteration on BLTL Trees

A key challenge in solving Problem 2 is that there are infinite belief states B_k. However, initial belief B_0 is always uniform, and there are a finite number of impulses $a_k \in \tilde{A}$. Notably, there are 2^Q possible observations $o_k \in \mathscr{O}$, which is much smaller than the number of interaction sequences of length $T(\varphi)$. Thus, at any iteration k, there are a *finite* number of possible next belief states B_{k+1}.

Hence, we solve Problem 2 in a receding horizon fashion, where at iteration k we construct a policy tree [3] starting at initial belief B_k. Each node in the tree is an impulse a_k, leaves are observations o_k, and subsequent nodes are actions a_{k+1} etc. Using value iteration on such policy trees with a horizon of H actions, we can select an optimal sequence of impulses a_k to a_{k+H}, and enact the plan depending on observations o_k. Finally, at belief node B_{k+H} we construct a new tree and optimal policy, which avoids optimizing over all possible belief vectors in $\tilde{B}$.

4.4 Size of BLTL trees

Satisfaction bitvectors provide significant reduction in the number of BLTL trees. The set of policy trees with $|\text{Act}^c|$ actions, $|\mathscr{O}|$ observations, and a horizon H has size $|\text{Act}^c|^{\frac{|\mathscr{O}|^H-1}{|\mathscr{O}|-1}}$ trees [3]. We first calculate the number of BLTL trees. The size of $\tilde{A}$, the set of all impulses, is $|\tilde{A}| = Q = \sum_i R_i$ consisting of all possible modes R_i for each model i. There are also 2^Q total observation bitvectors in $\mathscr{O}$, leading to

$$|\tilde{A}|^{\frac{|\mathscr{O}|^H-1}{|\mathscr{O}|-1}} = Q^{\frac{2^{QH}-1}{2^Q-1}} \tag{10}$$

trees. We now consider the number of trees if we used impulse actions, but observations were individual interaction histories of length $T(\varphi)$. Since the AS only acts with an impulse, we need not look at all possible actions in the interaction sequence, but rather just look at all possible state trajectories. Let the set of all observation histories of duration $T(\varphi)$ to be $\tilde{H}_{T(\varphi)}$, where $|\tilde{H}_{T(\varphi)}| = |\mathbf{S}|^{T(\varphi)}$. Then, we must optimize over

$$|\tilde{A}|^{\frac{(|\mathbf{S}|^{T(\varphi)})^H-1}{|\mathbf{S}|^{T(\varphi)}-1}} = Q^{\frac{(|\mathbf{S}|^{T(\varphi)})^H-1}{|\mathbf{S}|^{T(\varphi)}-1}} \tag{11}$$

trees, which scales with the size of state space $|\mathbf{S}|$, impulse bound $T(\varphi)$, and horizon H. Notably, the BLTL tree size with bitvectors *only* scales with the size of a smaller set of temporal logic formulae and horizon H. In Section 6, we show how a concise set of BLTL trees allows us to solve otherwise intractable problems.

4.5 Analysis

Theorem 1 (convergence). *Suppose that $(\mathscr{M}_1, \varphi_1)$ is the ground-truth agent-intent model (without loss of generality, modulo permutation of indices of candidate mod-*

els). Furthermore, suppose there is some $a \in \tilde{A}$ such that each $j \neq 1$,

$$\Pr(o \mid a, \mathscr{M}_1) \neq \Pr(o \mid a, \mathscr{M}_j) \tag{12}$$

for some observation o. Then, for each optimal policy ξ that solves Problem 2 with $\alpha = 0$, there is a time τ such that for all time $k \geq \tau$, the belief vector B_k has maximum value in the first position and no others, i.e., $B_k(1) > B_k(j)$ for $j \neq 1$.

Proof (sketch). Because the agent-intent model $(\mathscr{M}_1, \varphi_1)$ is the ground-truth by hypothesis, the true probability density function according to which observations are sampled is $\Pr(o \mid a, \mathscr{M}_1)$ for each abstract action a. As such, (12) implies that there is an abstract action such that every other candidate agent-intent model would yield observation vectors that differ in their expected frequency compared to the ground-truth (i.e., model 1). If $\alpha = 0$ in (9) (zero control cost), then from (8) it follows that the belief will tend to accumulate mass in position 1, i.e., for the ground-truth.

When the ground-truth stochastic dynamics or behavior specification is not in the set of candidate agent models, we may want the belief distribution to indicate when there is no match. Intuitively, the fitting process of belief states attempts to fit the distributions of satisfaction bitvectors per agent-intent model to the data. Thus, if none of the candidates is a good match, the belief will remain with high entropy and the AS can enact a conservative control policy. Due to space constraints, we provide rigorous full proofs of several relevant theorems, including general cases where $\alpha \neq 0$, in an online supplement [1].

5 Example Scenarios

In subsection 5.1, we provide BLTL formulae for adversarial car-following. Then, in 5.2, we show our approach extends to real SDD driving data. Finally, in 5.3.1, we introduce control costs that decrease as a robot gains certainty over human models, serving as a key contribution for modeling dynamic human-robot cooperation.

5.1 Robotic car-following BLTL specifications:

A robot transporting medical supplies must make subtle, costly route deviations to discern the nature of followers (Figure 1(b)). Follower models are referred to as `pursuant`, z`-bound` (for a surveillance car), or `benign` for a civilian. The robot's lane occupancy in one of X total lanes is denoted by system variables $\mathscr{Y} = \{C_1, \ldots, C_X\}$. Likewise, the follower's lane is denoted by environment variables $\mathscr{X} = \{F_1, \ldots, F_X\}$, leading to overall state $s_t = [C_{x,t} F_{x,t}]$ at time t. The z`-bound` car represents surveillance behavior since it must always stay within z lanes of the robot, yet is allowed the flexibility of T_z time steps to do so. We expand the BLTL formulae φ from [5] by adding control costs and more surveillance cars:

1. $\varphi_{\text{benign}} =$ `True`, i.e., civilian cars have no temporal logic constraint.
2. $\varphi_{z\text{-bound}} = \bigwedge_{x \in \{1,\ldots,X\}} \square \left(C_x \implies \diamondsuit_{[0,T_z]} F_{x-z} \vee F_{x-z+1} \vee \cdots \vee F_{x+z} \right)$
3. $\varphi_{\text{pursuant}} = \bigwedge_{x \in \{1,\ldots,X\}} \square \left(C_x \implies \diamondsuit_{[0,T_{\texttt{pursuant}}]} F_x \right)$ with time bound $T_{\texttt{pursuant}}$

[1] available at https://asl.stanford.edu/publications or directly,
https://asl.stanford.edu/wp-content/papercite-data/pdf/Chinchali.Livingston.Pavone.ISRR17.pdf

The robot can probe the follower by changing lanes which incurs a lane-deviation and fuel cost of $c(a_t) = 1$. Staying in the same lane incurs no cost. By proactively changing lanes, the robot learns if adversarial cars will eventually follow it to satisfy formulae of the form 2 and 3.

5.2 Stanford Drone Dataset (SDD)

The SDD [15] is a series of trajectories of bikers, pedestrians, and low-speed service carts that navigate crowded scenes on Stanford University's campus. Speed limits and restrictions on outside traffic make the campus a prime testing ground for autonomous vehicles. We consider how low-speed robotic carts might merge into crowded roundabouts such as in Figure 1(a) if they were allowed to *proactively* signal their merging intent. We note that potential cart-pedestrian accidents are only *minorly injurious* due to speed restrictions. Indeed, the SDD shows carts aggressively approaching pedestrians and the annotations and students alike jokingly refer to the roundabout in Figure 1(a) as the *death circle* only since it has *not* caused major accidents. Thus, a robot can plan rich, proactive merging strategies.

We performed extensive quality-control of over 69 GB of SDD data to identify trajectories where pedestrians and carts respond to rapidly changing traffic densities and intra-agent distances to merge or cross at opportune times (Figure 2).

5.3 Car-merging BLTL specifications

Let $\tilde{A} = \{s, u, n\}$ denote the cart's *safe* , *unsafe* , and *no signal* control actions, respectively. Cross traffic density $\rho \in \mathbb{R}$ is defined as the number of pedestrians per frame that travel perpendicular to the cart's merging direction. State $\mathbf{S} = [\rho, d, v]$ incorporates traffic density ρ, the cart's speed $v \in \mathbb{R}$, and its distance to a closest pedestrian $d \in \mathbb{R}$, as exemplified in Figure 2(b). Pedestrian models are denoted by $m \in \{cautious\ , daring\ \}$. Boolean variables $\tilde{\rho} \in \{h, l\}$ indicate if traffic is heavy ($h = \rho > \rho_0$) or light ($l = \rho < \rho_0$), where ρ_0 is a density threshold.

Given a traffic density $\tilde{\rho}$ and safety indication a, a pedestrian of model m will not cross for a time $T^m_{\tilde{\rho},a}$ to safely assess their surroundings, and after may cross based on their internal risk profile. Such behavior is captured by formula $\varphi^m_{\tilde{\rho},a}$ and an example for a *daring* pedestrian during heavy traffic ($\rho > \rho_0$) after the robot indicates *safe* is:

$$\varphi^{daring}_{h,s} = \square[\underbrace{(\rho > \rho_0)}_{\text{heavy}} \wedge \underbrace{safe}_{\text{impulse}}\ ] \implies \underbrace{\neg\ cross\ \mathbf{U}_{[0, T^{daring}_{h,s}\ ]}\ \texttt{True}}_{\text{can only cross after wait-time}} \tag{13}$$

The formula for a model m human covers all traffic scenarios $\tilde{\rho}$ and signals a:

$$\varphi^m = \bigwedge_{a \in \tilde{A}} \bigwedge_{\tilde{\rho} \in \{h,l\}} \varphi^m_{\tilde{\rho},a} \tag{14}$$

Crucially, BLTL time-bounds $T^m_{\tilde{\rho},a}$ allow the robot to differentiate models based on their crossing probabilities. For example, $T^{daring} < T^{cautious}$ regardless of signal since *daring* pedestrians deliberate for shorter times. Further, pedestrians wait longer if *unsafe* is indicated or traffic is light since the robot may merge. The robot waits a decision interval $T(\varphi) > T^m_{\tilde{\rho},a}$ for any model m, traffic condition $\tilde{\rho}$, and safety signal a to assess the pedestrian's response. Suppose the agent indicated *safe* dur-

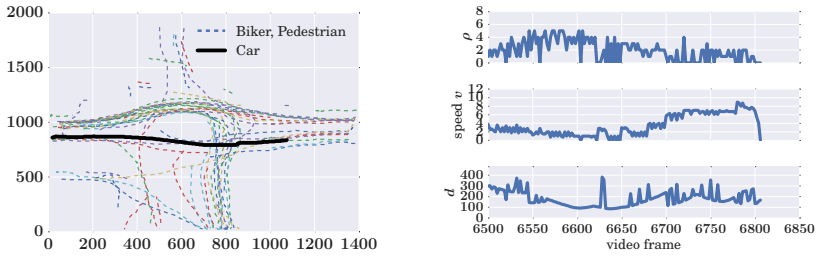

Fig. 2: A cart (bold) rapidly accelerates to merge when traffic ρ subsides after frame 6700.

ing very heavy traffic since it could not merge, yet it observed $\neg$ *cross* after $T(\varphi)$. Since the daring pedestrian is only constrained to not cross for $T_{h,s}^{daring} << T(\varphi)$, the probability of observing $\neg$ *cross* in a long interval might indicate a *cautious* model.

5.3.1 Robot behavior and dynamic control costs

In the following specifications, density and distance thresholds ρ_0, d_0, and speed multipliers M,L are mined from data [2]. We can learn such parameters automatically by contrasting velocity and traffic distributions during merging and steady-driving scenarios to find separating thresholds. Formulae 1 and 2 capture scenarios like Figure 2, where a cart cannot signal *safe* as it attempts to merge:

1. $\varphi^{slow} = \Box[\underbrace{(\rho > \rho_0)}_{\text{cross-traffic}} \wedge \underbrace{(d < d_0)}_{\text{biker close}} \wedge \underbrace{(v = v_0)}_{\text{speed}}] \implies \Diamond_{[0,T^{slow}]}[\underbrace{(v \leq \frac{v_0}{M})}_{\text{decelerate}}]$
2. $\varphi^{merge} = \Box[\underbrace{(\rho < \rho_0)}_{\text{light traffic}} \wedge \underbrace{(d > d_0)}_{\text{biker far}} \wedge (v = v_0)] \implies \Diamond_{[0,T^{merge}]}[\underbrace{(v \geq Lv_0)}_{\text{accelerate}} \wedge \underbrace{\neg\, safe}_{\text{disallow cross}}]$

Control costs capture the risk of a worst-case scenario where the cart causes an accident *after* indicating *safe* , so $c(s,B_k) > c(u,B_k) > c(n,B_k)$ for all k. Since the risk may decrease as the agent is more certain about the true pedestrian model, we also have $c(a,B_k) \propto H(B_k)$ for any action a.

6 Simulation results

In subsection 6.1, we show how dimensionality reduction allows us to solve an otherwise intractable belief MDP using value iteration. Then, in 6.2, we leverage advances in deep RL to scale our framework to larger problems with several competing models or cases where the ground truth model is not in the candidate set.

6.1 Value iteration tree results

Figure 3 shows a proactive car-merging strategy solved by BLTL tree value iteration. As introduced in Section 5.3.1, control costs capture model uncertainty and risk probabilities, so $c(a,B_k) = \frac{c_0(a)}{2}(1 + |\frac{H(B_k)}{H(B_0)}|)$ where $c_o(\text{safe}) = 0.40$, $c_0(\text{unsafe}) =$

[2] SDD provides annotations in terms of video frames and pixel distances, without calibration data. As such, recovering metric distances was infeasible, so we omit the values of these parameters.

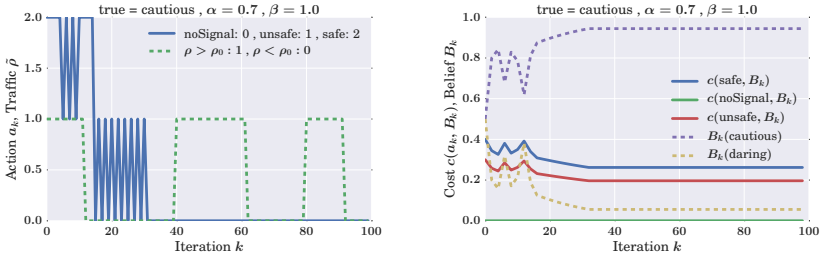

Fig. 3: (Left) In car-merging, the cart repeatedly signals *unsafe* to merge only after traffic ρ subsides. (Right) Control costs decrease as the cart quickly identifies the true cautious pedestrian.

0.30, and $c_0(\text{no signal}) = 0$ [3]. The cart initially chooses the high-cost, but most informative $safe$ action when traffic ρ is heavy and precludes merging. As the traffic subsides, the cart chooses the lower cost, but still informative $unsafe$ signal since it can now merge. Belief B_k indicates we correctly identify the true cautious pedestrian model and cost of informative controls decays over time as we increase model confidence (Figure 3(b)). Eventually, the agent chooses the zero cost *no signal* action once it has high model confidence and has already merged.

Using bitvector observations instead of individual interaction histories allows tractable value iteration. Even a simple car-following scenario with 4 lanes and time-bound of $T(\varphi) = 5$ steps would have $|\mathbf{S}| = 4^2 = 16$ joint robot-follower lane occupancies and $4^5 = 1024$ possible trajectories for the robot alone. If we consider only 3 BLTL formulae for pursuit, surveillance, or civilian behavior, 3 possible control actions $\tilde{A} = \{\text{left}, \text{right}, \text{stay}\}$, and a horizon of $H = 2$ impulses, there would be an integer overflow number of trees using interaction histories on a 64-bit computer. Notably, with bitvector observations, we only have a tractable 243 trees, since the observation space does *not* scale with the lane count or $T(\varphi)$.

6.2 Reinforcement Learning (RL) results

For complex problems with several candidates or an unanticipated model, enumerating even *high-level* observations and their probabilities is infeasible. Thus, we train an AS to tradeoff costly exploration with exploitation of the most likely model, *without* explicitly knowing model observation probabilities. Such a setting is a hallmark of RL, so we train an RL agent in a series of training episodes of K iterations. Each episode starts from a uniform belief B_0 and at step k, the agent chooses informative probe a_k based on current belief state B_k using parametrized control policy $\pi_\theta(B_k)$. The environment generates observations o_k under true model $\mathscr{M}_1$ and provides a new belief vector B_{k+1} and reward r_{k+1} to the agent, since the agent cannot compute the belief update itself without model probabilities.

We developed simulators for both examples using the *openAI gym* framework [4]. We used Google's Tensorflow [1] to learn a stochastic control policy using the Actor-Critic (AC) RL learning algorithm [10], where the policy $\pi_\theta(B_k)$ is encoded

[3] This is just one representative example of control costs allowed by our general framework.

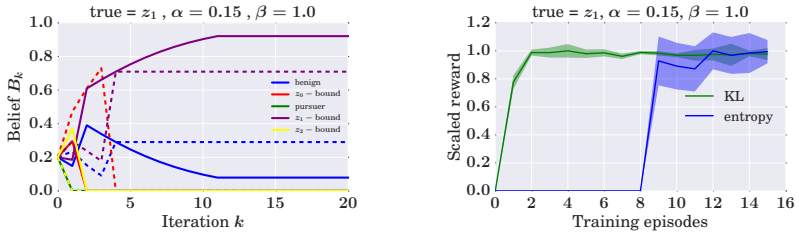

Fig. 4: (Left) Both KL and entropy rewards in the RL setting lead to correct model identification, but the KL trained policy (solid) identifies true model z_1 with higher certainty. (Right) Normalized RL learning curves for both reward functions indicates KL converges faster with lower variance.

in a neural network with parameters θ of 1 hidden layer of 50 units. Figure 4(b) illustrates model convergence, where the shaded area shows the variance of test episode rewards when the network policy is paused periodically to evaluate learning.

6.3 RL reward structures:

In addition to the entropy based reward from the belief MDP setting (Eqn. 9), we can formulate a reward that penalizes the KL divergence between the true model "one-hot" vector $\bar{B} = [1,0,\ldots 0]$ and the current belief. The following reward is only appropriate in the RL scenario where the environment simulator knows the true model and incentivizes the agent to learn the ground-truth:

$$r_k^{KL}(B_k, a_k, B_{k+1}) = -\alpha c(a_k, B_k) - \beta KL(B_{k+1}, \bar{B}), \tag{15}$$

where KL is the Kullback-Leibler divergence, $c : \tilde{A} \times \tilde{B} \to \mathbb{R}$ is the control cost function, and $\alpha > 0, \beta > 0$. Since the KL divergence is always positive, we weight by $-\beta$ to penalize excessive differences between B_k and $\bar{B}$.

Interestingly, for a wide spectrum of weights α, β, the policy learned under the KL reward converged faster than the entropy reward for the same experimental settings (Figure 4(b)). Further, in a single test episode of K iterations, the KL-learned policy led the agent to identify the true model with more certainty (Figure 4(a)). Intuitively, if the expected future entropy reduction is lower than the cost of informative probes, the agent will stop probing but incur zero future reward since the belief vector will saturate. However, the KL reward converges better since it *continually* penalizes KL divergence between the current belief and true distribution throughout the episode, incentivizing longer exploration to reduce uncertainty.

7 Conclusion

In this paper, we couple formal methods with data-driven learning to provide a tractable framework for proactive decision making. Formal methods are used to extract meaningful symbolic interaction templates from complex interaction sequences, such as traces of real human driving data in the SDD. Leveraging advances in deep RL, we then synthesize information-seeking controllers and provide a theoretical analysis of their ability to distinguish models.

Future work centers on developing an experimental car-merging testbed. We plan to conduct user studies where a simulated autonomous cart signals its merging intent using a ProDM scheme and human subjects deliberate on whether to cross, allowing us to directly determine human risk profiles when *explicitly probed*. Then, such risk profiles can be combined with studies on the financial consequences of minor accidents to select control costs that capture mutual human-robot trust. To solve problems with a larger spectrum of agent types, we plan to use shared generative models, such as pre-trained neural networks that capture general driver behavior.

As robots cooperate with humans on increasingly complex tasks, techniques that distill a continuum of high-dimensional interaction sequences into core essential templates of interaction will be evermore indispensable. Such a holistic approach to robot task planning may one day allow robots to effectively cooperate with humans in diverse settings ranging from factory assembly lines to freeways.

References

1. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of the OSDI 2016. Savannah, Georgia, USA*, 2016.
2. C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
3. D. Braziunas. POMDP solution methods: a survey. Technical report, Department of Computer Science, University of Toronto, 2003.
4. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
5. S. P. Chinchali, S. C. Livingston, M. Pavone, and J. W. Burdick. Simultaneous model identification and task satisfaction in the presence of temporal logic constraints. In *ICRA*, 2016.
6. P. J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multi-agent settings. *J. Artif. Intell. Res.(JAIR)*, 24:49–79, 2005.
7. S. Javdani, S. S. Srinivasa, and J. A. Bagnell. Shared autonomy via hindsight optimization. In *RSS*, 2015.
8. A. Jones, M. Schwager, and C. Belta. Information-guided persistent monitoring under temporal logic constraints. In *ACC, 2015*, pages 1911–1916. IEEE, 2015.
9. W. Knight. New self-driving car tells pedestrians when it's safe to cross the street. *MIT Technology Review*, 2016.
10. V. Konda and J. Tsitsiklis. Actor-critic algorithms. In *NIPS*, volume 13, pages 1008–1014, 1999.
11. O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.
12. T.-H. D. Nguyen, D. Hsu, W.-S. Lee, T.-Y. Leong, L. P. Kaelbling, T. Lozano-Perez, and A. H. Grant. Capir: Collaborative action planning with intention recognition. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.
13. C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.
14. V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia. Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 239–248. ACM, 2015.
15. A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, pages 549–565. Springer, 2016.
16. D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan. Information gathering actions over human internal state. In *IROS*, 2016.
17. P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proceedings of IROS*, 2010.
18. T. Wongpiromsarn and E. Frazzoli. Control of probabilistic systems under dynamic, partially known environments with temporal logic specifications. In *Proceedings of CDC*, pages 7644–7651, December 2012.