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

Journal Title
XX(X):1–22

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Sandeep P. Chinchali[1], Scott C. Livingston[2], Mo Chen[1], and Marco Pavone[1]

Abstract

The operation of today's robots entails interactions with humans, e.g., in autonomous driving amidst human-driven vehicles. To effectively do so, robots must proactively decode the intent of humans and concurrently leverage this knowledge for safe, cooperative task satisfaction—a problem we refer to as proactive decision making. However, simultaneous intent decoding and robotic control requires reasoning over several possible human behavioral models, resulting in high-dimensional state trajectories. In this paper, we address the proactive decision making problem using a novel combination of formal methods, control, 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 maintains a belief distribution over models of human behavior, and proactively plans informative actions. After showing several desirable theoretical properties, we apply our framework to a dataset of humans driving in crowded merging scenarios. For it, temporal logic models are generated and used to synthesize control strategies using tree-based value iteration and deep reinforcement learning (RL). Additionally, we illustrate how data-driven models of human responses to informative robot probes, such as from generative models like Conditional Variational Autoencoders (CVAEs), can be clustered with formal specifications. Results from simulated self-driving car scenarios demonstrate that data-driven strategies enable safe interaction, correct model identification, and significant dimensionality reduction.

1 Introduction

Data-driven learning is a key ingredient of modern autonomous systems (AS). However, in many practical settings, from surgical robots to autonomous cars, this learning and control must occur while seamlessly interacting with other agents. This in turn requires understanding the agents' 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 started to focus significant attention on *proactive* intent decoding and decision making Knight (2016); Sadigh et al. (2016). Examples include autonomous cars that gently nudge into adjacent lanes to discern the driving style of nearby drivers for lane-merging Sadigh et al. (2016) or use large signs to proactively signal when pedestrians can safely cross at intersections Knight (2016).

An important challenge of *proactive* decision making coupled with *concurrent* robotic control is that the resulting decision-making problem involves interaction with another agent. 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 a combination of formal methods, control, 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 Trautman and Krause (2010), this gap is partially bridged by modeling interdependency of

[1] Stanford University, Stanford CA 94305, USA
[2] rerobots, Inc., Walnut CA 91789, USA

Corresponding author:
Marco Pavone, 496 Lomita Mall, Rm. 261 Stanford, CA 94305
Email: pavone@stanford.edu

human-robot planning using Gaussian processes. Recently, Sadigh et al. (2016) 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 Madani et al. (1999); Papadimitriou and Tsitsiklis (1987). Relevant prior work using POMDPs includes hindsight optimization for grasping Javdani et al. (2015), interactive POMDPs (I-POMDPs) Gmytrasiewicz and Doshi (2005), and goal decomposition approaches Nguyen et al. (2011). Recent work on temporal logic models highlights their ability to capture safety constraints and high-level interactions Jones et al. (2015); Raman et al. (2015); Wongpiromsarn and Frazzoli (2012). In particular, Wongpiromsarn and Frazzoli (2012) is especially relevant since it casts human-robot interaction as an *adversarial Markov Decision Process (aMDP)* where a robot must synthesize a strategy to maximize the probability of satisfying a formal interaction specification by reasoning over several possible environment (human) behavioral modes, which quickly becomes intractable for long-interaction horizons with a plethora of plausible human models. Unlike Wongpiromsarn and Frazzoli (2012), which simply maximizes specification satisfaction probabilities, we instead use an explicit reward signal in the aMDP formulation to incentize information-seeking, *proactive* decision making. Further, 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: This paper addresses control problems with objectives that depend on interaction with an unknown, possibly adversarial agent. Though this setting appears to have high dimensionality, we show how to 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) Robicquet et al. (2016), 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. Lastly, our approach is demonstrated with generative models applied to simulations of two cars in a highway lane merging scenario.

A preliminary version of this work appeared at the 2017 International Symposium on Robotics Research (ISRR). In this revised and extended version, we provide the following additional contributions: (i) a new generative modeling case study of highway lane merging, (ii) characterization of optimal policies in special cases, (iii) empirical study of exploration-exploitation trade-off for an RL agent, and (iv) proofs of all theoretical results.

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 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 Robicquet et al. (2016). Such examples of human-robot interaction are already starting to be addressed in industry. For example, autonomous car startup *drive.AI* Knight (2016) has proposed to equip vehicles with large signs that indicate when pedestrians can safely cross at intersections. Inspired by *drive.AI*'s proposal, the robotic car in our example can *proactively* instruct pedestrians to wait, safely cross, 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.

Example 3. Highway lane-merging. In Figure 1(c), a robotic car (red) must switch lanes with a human driven vehicle (green) within a short time span and length of a highway. To negotiate this complex interchange, the robot may accelerate to probe whether the human driver will slow down and yield to the robot or whether the human is aggressive, requiring the robot to slow down for the human. Thus, the robot must balance exploration of human driver style, which comes with a control cost on unsafe accelerations, with exploitation of its currently assumed model to smoothly change lanes.

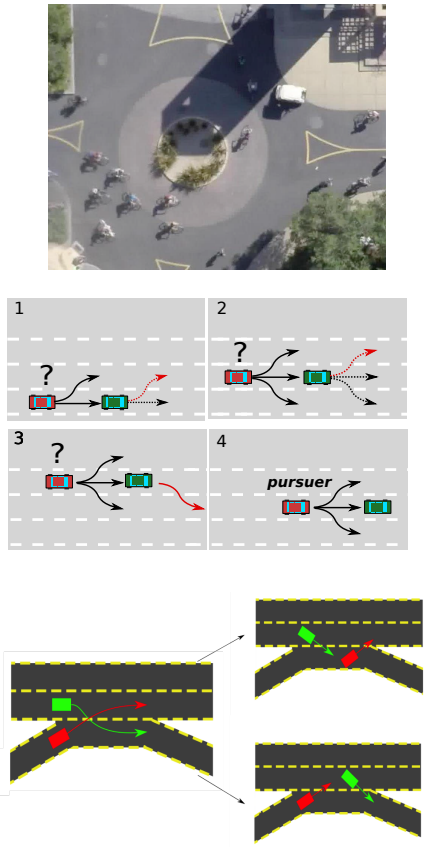

Figure 1. Examples of Proactive Decision Making: (a, top) A cooperative scenario, from the Stanford Drone Dataset Robicquet et al. (2016), shows how cars must nudge into crowded roundabouts. (b, middle) An adversarial scenario, inspired by Chinchali et al. (2016), shows how a green robotic car must safely swerve lanes to determine if it is being pursued by a red adversarial car. (c, bottom) A robotic vehicle (red) must proactively probe whether a human driven vehicle (green) will allow it to swap lanes on a highway on-ramp.

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 formal specification language used throughout the paper, Bounded Linear Temporal Logic (BLTL), is introduced. To model probabilistic interaction between a robot and human, we introduce an *adversarial* Markov Decision Process (MDP) with labelings that allow it to be evaluated with respect to a formal specification. Finally, we formulate Problem 1, which concerns finding an AS control policy to disambiguate between several candidate human models defined as adversarial MDPs with associated formal specifications.

Consider a set of m agents operating in discrete time. Let $\mathcal{A} = \{1, \ldots, m\}$ 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.

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

$$\mathcal{H}_T(t) = \big[(s_t, \mathbf{a}_t), \ldots, (s_{t+T-1}, \mathbf{a}_{t+T-1}), s_{t+T}\big],$$

where $\mathcal{H}_T(t)$ is said to begin at discrete time t 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 $\mathcal{H}_T(0)$ starting at $t = 0$ is called an *interaction history* and is also denoted as $\mathcal{H}_T$.

As an example, in the car-following scenario, the set of agents comprises the robot and the follower, states are the lane occupancies of cars, and actions are decisions to stay with the current trajectory or probe the follower by moving to an adjacent lane.

3.1 Bounded Linear-time Temporal Logic (BLTL)

We now introduce formal specifications that are used to model-check and reason about safety or high-level intent encoded in interaction sequences. In this paper, formal specifications of interaction sequences are defined for *finite* durations that are well-suited to information gathering tasks. We employ bounded linear-time temporal logic (BLTL), which was introduced in Chinchali et al. (2016) and summarized below. It is a fragment of metric temporal logic (MTL), a general specification language for time-dependent properties originally defined by Koymans (1990). An introduction to basic concepts, including model checking MDPs, can be found in the book by Baier and Katoen (2008). 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 $\mathcal{Y} = \{C_x\}_{x=1}^{X}$ to indicate the robot occupies lane x of X total lanes.

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

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

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.

As an example, 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:

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

3.2 High-level Agent Intent Models

We now introduce a framework to capture probabilistic interactions between a robot and human, which can be model-checked using formal specifications. Henceforth, we treat the case of two interacting agents – an autonomous system that we control (denoted by `AS`), and an uncontrolled *adversary* denoted by `adv`. Our framework is general and the generic term adversary is used to denote an uncontrolled, human agent. It can be either adversarial to the AS or cooperative, but we refer to it as an adversary to address a conservative case where the human agent may not want to readily reveal its true intent to the robot. To reason about the stochastic interaction of an AS and adversary, we use *labeled adversarial Markov Decision Processes (aMDPs)*, which are defined similarly as in Wongpiromsarn and Frazzoli (2012) in Definition 2.

Definition 2. Labeled adversarial MDP (aMDP). A *labeled adversarial MDP* $\mathcal{M}$ is a tuple $(\mathbf{S}, \text{Init}, \text{Act}^c, \text{Act}^u, \mathbf{P}, \Pi, L)$, where $\mathbf{S}$ is a finite set of states, $\text{Init} \subseteq \mathbf{S}$ 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 $\mathbf{P} : \mathbf{S} \times \text{Act}^c \times \text{Act}^u \times \mathbf{S} \to [0, 1]$ defines transition probabilities where for each state $s \in \mathbf{S}$, $a \in \text{Act}^c(s)$, and $b \in \text{Act}^u(s)$, $\sum_{s' \in \mathbf{S}} \mathbf{P}(s, a, b, s') = 1$.

We now introduce a key assumption that allows both the AS and adversary to take actions from every state to allow for rich control policies.

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

Intuitively, this assumption stipulates that no dead-ends exist, i.e., from every state, both the controlled robot and the adversary have at least one possible action. For any state $s \in \mathbf{S}$, the actions in $\text{Act}^c(s) \cup \text{Act}^u(s)$ are said to be *enabled*.

3.3 Probability of Satisfaction and Agent-intent Models

Now, we quantify the probability that a stochastic interaction sequence satisfies each plausible human intent, which allows the AS to infer the most likely human intent from its observations. Let $\mathcal{M}$ be a labeled adversarial MDP. A strategy is a partial function from an interaction history to exactly one of the two action sets associated with $\mathcal{M}$: 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

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

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

$$\mathrm{Traces}(\mathcal{M},T,\pi,\mu)= \\ \big\{\sigma\in\Sigma^{T+1} \mid \exists \mathcal{H}_T\in\mathrm{Hist}(\mathcal{M},T,\pi,\mu): \\ \forall\tau:0\le\tau\le T\wedge \sigma a_\tau=L(s_t)\big\}, \quad (3)$$

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 $\sigma\in\mathrm{Traces}(\mathcal{M},T,\pi,\mu)$ in (3) on some $\mathcal{H}_T\in\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$ can be generalized to show there is a function $\mathcal{L}$ from $\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$ onto $\mathrm{Traces}(\mathcal{M},T,\pi,\mu)$ consistent with the comprehension in (3), i.e., such that for all $\mathcal{H}_T\in\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$, $\mathcal{L}(\mathcal{H}_T)\in\mathrm{Traces}(\mathcal{M},T,\pi,\mu)$ and $\mathcal{H}_T$ realizes the existential quantification in (3).

Let φ be a BLTL formula defined in terms of atomic propositions from Π. Recall that there is a minimal bound $T(\varphi)$ such that for any $T\ge T(\varphi)$ 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 $\mathcal{H}_T\models\varphi$ to be true if and only if $\mathcal{L}(\mathcal{H}_T)\models\varphi$, which indeed is well-defined for $T\ge T(\varphi)$ because $\mathrm{Traces}(\mathcal{M},T,\pi,\mu)\subseteq\Sigma^{T+1}$.

The adversarial MDP $\mathcal{M}$ has finite state and action sets, so for $T<\infty$, it follows that the set $\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$ is finite for any policies π and μ. Entirely similar observations hold for $\mathrm{Hist}(\mathcal{M},T)$ and $\mathrm{Traces}(\mathcal{M},T,\pi,\mu)$.

Each interaction history $\mathcal{H}_T\in\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$ defines a finite sequence of states and actions. From (2), it follows that $\mathcal{H}_T$ is equivalent to a finite sequence of state-action-state triples, each of which is a transition of $\mathcal{M}$. Recall that the probability of each transition is defined by $\mathbf{P}$. Computing the product of these probabilities for each element of $\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$ and normalizing provides a probability mass function over $\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$. For sufficiently large T, it can be decided for each element of $\mathrm{Hist}(\mathcal{M},T,\pi,\mu)$ whether it satisfies the BLTL formula φ, hence we define

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

as the probability of satisfying φ when AS strategies and adversarial strategies are applied to $\mathcal{M}$. When Init is a singleton, we write $P_{\mathcal{M},\pi,\mu}(s\models\varphi)$.

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

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

The following assumption states that the adversary, if following model $(\mathcal{M},\varphi)$, 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 since otherwise its specification is not an accurate description for the adversary's true behavior. To allow for *unbiased* measurement uncertainty, we only assume satisfaction in probability.

Assumption 2. Strategy matches specification. *Let $(\mathcal{M},\varphi)$ be an agent-intent model. For any $s\in$ Init, the adversarial strategy is an element of*

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

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

3.4 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 and an adversary, referred to as adv. Our framework is general and allows the uncontrolled adversary to be either cooperative or adversarial. The AS must find a control policy π_{AS} that maximizes reward R despite possible interference from the adversary enacting controller π_{adv}. Reward function $R:\mathcal{H}_T\mapsto\mathbb{R}$ defines a scalar reward over interaction history $\mathcal{H}_T$ which incentivizes the AS to learn about the true model $(\mathcal{M}_j,\varphi_j)$ 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* $\{(\mathcal{M}_1, \varphi_1), \ldots, (\mathcal{M}_N, \varphi_N)\}$*, find strategy* π^*_{AS} *such that*

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

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

In terms of the car-following example, R is a weighted sum of fuel and lane deviation cost coupled with a metric of information gain. The cost in R could also include distance to a destination, incentivizing goal-driven behavior. Specifications φ govern how different follower car models pursue the robot, while interaction sequences $\mathcal{H}_T$ are joint robot-follower lane trajectories. When the ground-truth model is not in the finite set of initial models $\{(\mathcal{M}_1, \varphi_1), \ldots, (\mathcal{M}_N, \varphi_N)\}$, the desired behavior of the AS strategy is to identify the closest behaving model to the observed behavior or, in extreme cases, declare that no assumed models are concordant with the observed behavior and enact a conservative control policy.

4 Dimensionality Reduction and Solution Algorithm

Problem 1 is typically computationally intractable for large-scale problems because, as a consequence of not knowing the ground-truth model index j, we must optimize over all possible agent-intent models $(\mathcal{M}_i, \varphi_i)$ and interaction sequences $\mathcal{H}_T$. In this section, we present, solve, and theoretically analyze a new problem based on dimensionality reductions from Problem 1.

4.1 Action and Observation Space Reductions

Figures 2 and 3 illustrate the key insights behind our dimensionality reduction approach in terms of the mathematical notation we now introduce.

4.1.1 Action Space: Request-response formulae

The first dimensionality reduction comes from using formal specifications to select a concise set of informative probes to reduce the action space of proactive decision making. For a special form of BLTL formulae that is defined below, we are able to reduce the size of the action space of adversarial MDPs. Let $\{(\mathcal{M}_1, \varphi_1), \ldots, (\mathcal{M}_N, \varphi_N)\}$ be a set of agent-intent models (recall Definition 3 in Section 3.3).

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

$$\varphi_i = \bigwedge_{p \in \{1, \ldots, p_i\}} \left(\psi^{\text{req},p}_{AS,i} \implies \Diamond_{[0,T_p]} \psi^{\text{res},p}_{\text{adv},i} \right) = \bigwedge_{p \in \{1, \ldots, p_i\}} \varphi^p_i, \tag{5}$$

where each mode p represents an AS (robot) *probe* to agent i given by $\psi^{\text{req},p}_{AS,i}$, $\psi^{\text{res},p}_{\text{adv},i}$ is the *adversary's informative response*, and T_p 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. Deadlock situations, such as a pedestrian waiting infinitely long at an intersection, rarely persist indefinitely. Even if they do, AS, such as search and rescue drones, can employ a flexible approach where they move to a new search location after a timeout but can revisit target sites. 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_p. 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.

4.1.2 Observation Space: Satisfaction Bitvectors

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. Figure 2 illustrates the definition using Example 2 (adversarial car-pursuit).

Definition 5. Satisfaction bitvector observation. Consider N high-level agent intent models $M_i = (\mathcal{M}_i, \varphi_i)$ (Def. 3 in Section 3.3) where each specification φ_i has p_i BLTL request-response formulae. Define $Q = \sum_i p_i$, and $T = \max_i T(\varphi_i)$. The *satisfaction bitvector observation* o is a function that maps an interaction history $\mathcal{H}_T$ to a vector of dimension Q where the q-th element is 1 if $\mathcal{H}_T$ satisfies the qth interaction formula $\varphi_q = \varphi^p_i$ and is otherwise 0. Observation space $\mathcal{O} = \{o \mid o \in \{0,1\}^Q\}$ comprises 2^Q possible bitvectors. Written in terms of the indicator function $\mathbb{1}$,

$$o(\mathcal{H}_T) = [\mathbb{1}(\mathcal{H}_T \models \varphi^1_1), \ldots, \mathbb{1}(\mathcal{H}_T \models \varphi^{p_N}_N)]. \tag{6}$$

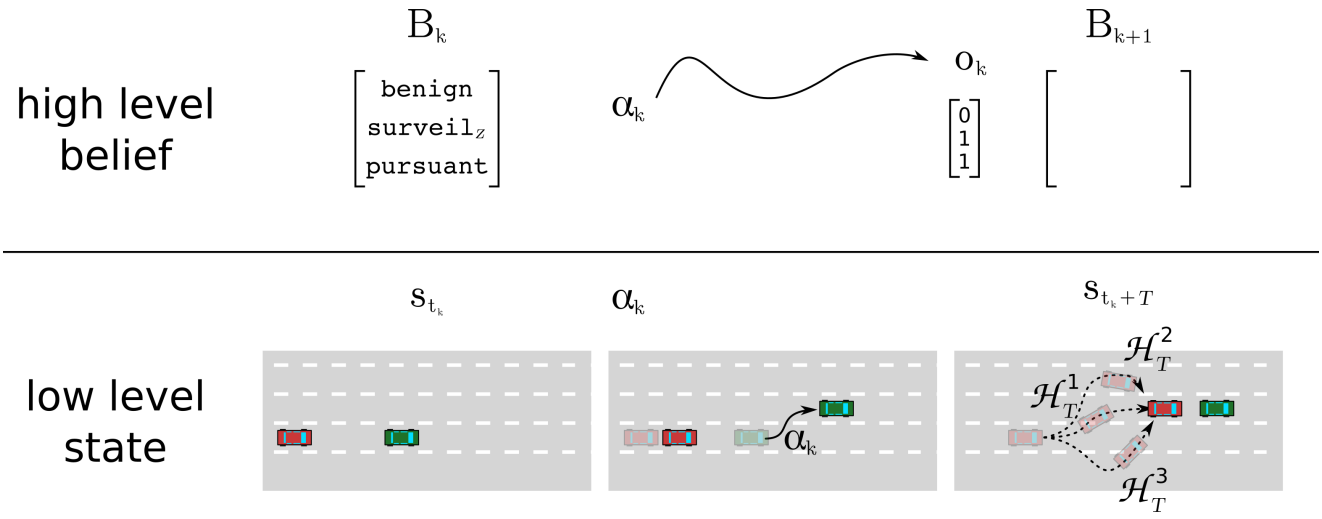

Figure 2. Illustration of Bitvector Observations: Our key algorithmic insight, illustrated for the car-following example, is to plan in the belief space over agent intent models (top panel) as opposed to the low-level state space (bottom panel). At the start of iteration k, where the low level state is s_{t_k}, the green robotic car probes the red follower's intent by swerving lanes with action α_k. Rather than optimize over all possible interaction histories $\mathcal{H}_T$ in the low-level state space that end with the follower pursuing the robot (bottom right), we simply encode the information in bitvector o_k. Concise bitvector observation o_k describes whether interaction history $\mathcal{H}_T$ adheres to each possible model's specification, which is then used to update the belief B_k over agent intents by solving Problem 2.

After BLTL impulse actions, satisfaction bitvector observations constitute the second key dimensionality reduction technique presented in this paper. Rather than reasoning about a continuum of interaction histories which may reside in a high-dimensional state space of adversarial MDPs, bitvector observations directly apply formal methods to distill the interaction into its essence, namely which high-level intents it is concordant with.

4.2 Reduced ProDM-BLTL Problem with Belief MDPs

Leveraging the previous definitions, we reduce the dimensionality 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 $M_B = (\tilde{B}_0, \tilde{B}, \tilde{A}, P_b, \tau_b)$ 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. $\tilde{A} = \{\psi_{AS,i}^{\text{req},p} \mid i \in \{1,\ldots,N\}, p \in \{1,\ldots,p_i\}\}$.
   Abstract (impulse) actions are controlled requests for any BLTL request-response formula. To reduce clutter in the notation, we will often use the symbol α to denote an element in $\tilde{A}$.
3. $P_b : \tilde{B} \times \tilde{A} \times \tilde{B} \to \mathbb{R}$
   The belief transition function P_b is defined by

$$\begin{aligned} P_b(B_k, \alpha_k, B_{k+1}) &= \Pr(B_{k+1} \mid B_k, \alpha_k) \\ &= \left(\sum_{o_k \in \mathcal{O}} \Pr(B_{k+1} \mid B_k, \alpha_k, o_k) \right) \\ &\quad \cdot \left(\sum_{i=1}^{N} \Pr(o_k \mid \alpha_k, \mathcal{M}_i) B_k(i) \right) \end{aligned}$$

   in which

$$\begin{aligned} &\Pr(o_k \mid \alpha_k, \mathcal{M}_i) = \\ &\quad \prod_{q=1}^{Q} \Big(o_k^q \Pr(\mathcal{H}_T(t_k) \models \varphi_q \mid \alpha_k, \mathcal{M}_i) \\ &\quad + (1 - o_k^q) \Pr(\mathcal{H}_T(t_k) \not\models \varphi_q \mid \alpha_k, \mathcal{M}_i) \Big), \end{aligned} \quad (7)$$

   where $o_k = (o_k^1, \ldots, o_k^Q) = o(\mathcal{H}_T(t_k))$ and φ_q are from Def. 5. Notice that $\Pr(o_k \mid \alpha_k, \mathcal{M}_i)$ 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 $(\mathcal{M}_i, \varphi_i)$.
4. The belief update function $\tau_b : \tilde{B} \times \tilde{A} \times \mathcal{O} \to \tilde{B}$ maps belief state B_k, action α_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 α_k under each competing model M_i, i.e., $B_{k+1} =$

$\tau_b(B_k, \alpha_k, o_k)$ where

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

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 R is a weighted sum of control cost and information gain. In this paper, we consider two different stage rewards. The first stage reward is based on entropy reduction in the belief vector:

$$R_k^H(B_k, \alpha_k, B_{k+1}) = -\beta_{\mathrm{C}} c(\alpha_k, B_k) + \beta_{\mathrm{I}}[H(B_k) - H(B_{k+1})]. \quad (9)$$

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

The second stage reward we consider is based on the Kullback-Leibler divergence, and is useful in situations, for example in simulation, when the true adversarial MDP model is known:

$$R_k^{\mathrm{KL}}(B_k, \alpha_k, B_{k+1}) = -\beta_{\mathrm{C}} c(\alpha_k, B_k) - \beta_{\mathrm{I}} \mathrm{KL}(B_{k+1}, \bar{B}), \quad (10)$$

where KL is the Kullback-Leibler divergence, and $\bar{B}$ is a belief vector with the element 1 in the position corresponding to the true adversarial MDP model, and 0 elsewhere. Without loss of generality, in this paper we will assume that the true adversarial MDP has an index of 1, so that $\bar{B}(1) = 1, \bar{B}(j) = 0$ for all $j = 2, 3, \ldots, N$. Since the KL divergence is always positive, we penalize differences between B_k and $\bar{B}$.

4.2.1 Intuition behind dimensionality reduction

Figure 2 illustrates the motivation behind the *belief MDP* approach by contrasting the high-level belief space and the possibly high-dimensional low-level state space in terms of the car-following example. Figure 3 depicts the general setting, where low-level state space featuring several possible interaction sequences is on the left, and it is juxtaposed with the simpler high-level belief space featuring a concise set of bitvector observations on the right. In general, a transition of M_B occurs in two parts starting at an iteration k (Figs. 2 and 3.). First, we start at a belief state B_k at time t_k when the fully observable adversarial MDP state is s_{t_k}. Note that here, we are using the notation t_k to denote the correspondence between time indices in the adverarial MDP $\mathcal{M}$, denoted t, and time indices in the belief MDP M_B, denoted k.

An action in the belief MDP α_k is selected, corresponding to a specific BLTL request-response formula for some model i and high-level mode p. By selecting this action, we assume model i is the true model and synthesize a strategy corresponding to adversarial MDP $\mathcal{M}_i$ and interaction formula φ_i, using Assumption 2 in Section 3.3.

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 $T(\varphi)$, $\mathcal{H}_T(t_k)$. The interaction sequence can then be mapped to a satisfaction bitvector observation $o_k = o(\mathcal{H}_T(t_k))$. Given current belief state B_k, action α_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.

4.2.2 Proactive Decision Making in Belief Space

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

Problem 2. Belief-ProDM: ProDM-BLTL with Belief MDP. *Let* $\{(\mathcal{M}_1, \varphi_1), \ldots, (\mathcal{M}_N, \varphi_N)\}$ *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*

$$\begin{aligned} \underset{\bar{\alpha}}{\textit{maximize}} \quad & \mathbb{E}\left(\sum_{k=0}^{K-1} \gamma^k R_k(B_k, \alpha_k, B_{k+1})\right) \\ \textit{subject to} \quad & \Pr(B_{k+1} \mid B_k, \alpha_k) = P_b(B_k, \alpha_k, B_{k+1}) \\ & \textit{for all } k = 0, \ldots, K-1 \end{aligned}$$

where $\bar{\alpha} = \{\alpha_k\}_{k=0}^K$, and the reward R_k can be chosen to be either R_k^H or R_k^{KL}.

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 $\alpha_k \in \tilde{A}$. Notably, there are 2^Q possible observations $o_k \in \mathcal{O}$, 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 Braziunas (2003) starting at initial belief B_k. The right of Figure 3 depicts such a policy tree. Each node in the tree is an impulse α_k, leaves are observations o_k, and subsequent nodes are actions α_{k+1} etc. Using value iteration on such policy trees

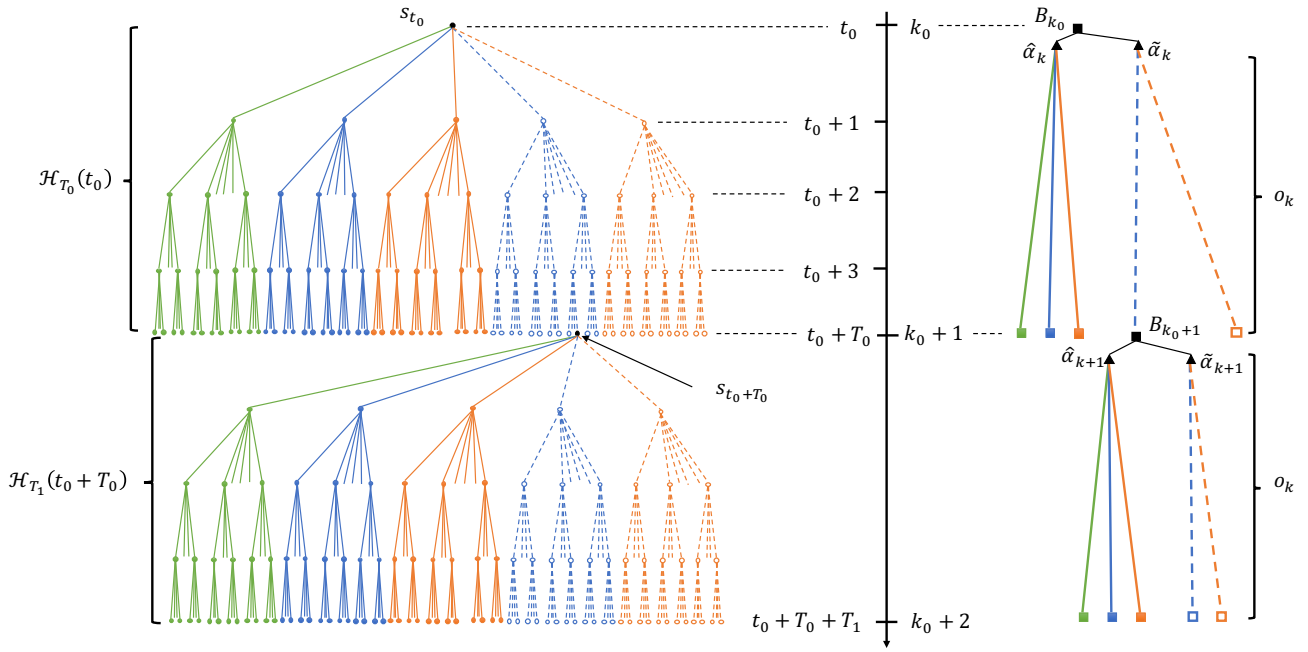

Figure 3. Illustration of dimensionality reduction through the belief MDP. (Left) Evolution of the possibly high-dimensional, low-level state s_t in continuous time. Edges with a common color indicate interaction histories $\mathcal{H}_T$ that correspond to the *same* bitvector observation o_k in terms of BLTL. (Right) Evolution of a Belief MDP in iterations k, each of BLTL time bound T_k. The key insight necessary for dimensionality reduction is that several interaction histories can be clustered into a smaller set of bitvectors o_k.

with a horizon of h actions, we can select an optimal sequence of impulses α_k to α_{k+h-1}, 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, as illustrated in Figure 3 and shown in this section. The set of policy trees with $|\,\mathrm{Act}^c\,|$ actions, $|\mathcal{O}|$ observations, and a horizon h has size $|\,\mathrm{Act}^c\,|^{\frac{|\mathcal{O}|^h-1}{|\mathcal{O}|-1}}$ trees Braziunas (2003).

We first calculate the number of BLTL trees with impulse actions and observation bitvectors. The size of $\tilde{A}$, the set of all impulses, is $|\tilde{A}| = Q = \sum_i p_i$ consisting of all possible modes p_i for each model i. There are also 2^Q total observation bitvectors in $\mathcal{O}$, leading to

$$|\tilde{A}|^{\frac{|\mathcal{O}|^h-1}{|\mathcal{O}|-1}} = Q^{\frac{2^{Qh}-1}{2^Q-1}} \tag{11}$$

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 $\tilde{\mathcal{H}}_{T(\varphi)}$, where $|\tilde{\mathcal{H}}_{T(\varphi)}| = |\mathbf{S}|^{T(\varphi)}$. Then, we must optimize over

$$|\tilde{A}|^{\frac{(|\mathbf{S}|^{T(\varphi)})^h-1}{|\mathbf{S}|^{T(\varphi)}-1}} = Q^{\frac{(|\mathbf{S}|^{T(\varphi)})^h-1}{|\mathbf{S}|^{T(\varphi)}-1}} \tag{12}$$

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. In particular, we show how we can solve the car-following problem with a small set of BLTL observation bitvector trees, but the same problem using observation histories would have led to an integer overflow (intractable) number of trees.

4.5 Analysis

In this section, we state the main theoretical results of this paper. Proofs can be found in the appendix. We first address the case of no control cost, given by control cost weight $\beta_C = 0$, in order to establish that the AS will converge to the ground truth model when it is not penalized for probing agent intent (Lemma 1 and Theorem 1). Then, to allow for strong theoretical guarantees, we consider a simplified problem setting in subsection 4.5.2 where the AS has a control cost penalty for probes given by $\beta_C > 0$ (Lemmas 2 and 3).

4.5.1 Convergence without control cost

Recall the definition of belief MDP from Section 4.2,

and the distinction between actions and abstract actions, as illustrated in Figure 3.

Lemma 1. Convergence to ground-truth. *Suppose that $(\mathcal{M}_1, \varphi_1)$ is the ground-truth agent-intent model. Furthermore, suppose there is some $\alpha \in \tilde{A}$ such that for each $j \neq 1$,*

$$\Pr(o \mid \alpha, \mathcal{M}_1) \neq \Pr(o \mid \alpha, \mathcal{M}_j) \tag{13}$$

for some observation o.

Let $\bar{\alpha}$ be any policy such that, for each $j \in \{1, \ldots, N\}$, infinitely often an abstract action α is selected such that for some satisfaction observation bitvector o,

$$\Pr(o \mid \alpha, \mathcal{M}_j) \neq \Pr(o \mid \alpha, \mathcal{M}_1). \tag{14}$$

Then, $\lim_{k\to\infty} B_k(1) = 1$ if and only if the policy $\bar{\alpha}$ is used.

Remark 1. Policy construction. *Note that it is possible to construct a policy with this property without knowing that the ground-truth model is $\mathcal{M}_1$ because the following stronger requirement can instead be used. Because $\Pr(o \mid \alpha, \mathcal{M}_j)$ can be computed for any $j \in \{1, \ldots, N\}$ (independently of which candidate model corresponds to the ground-truth), a modified form of (14) can be checked between each pair of models. There will be at least one candidate model that can take the role of $\mathcal{M}_1$ in (14). Thus, in general we can let $\bar{\bar{\alpha}}$ be a policy that selects infinitely often abstract actions that realize all of those inequalities (i.e., for all of which were found to be satisfiable).*

Theorem 1. Optimality with no control cost. *Any policy $\bar{\alpha}$ defined in Lemma 1 is optimal for Problem 2 when $\beta_C = 0$ and $\gamma = 1$, with an infinite time horizon, $K \to \infty$.*

4.5.2 Exploration-exploitation trade-off with control costs

For the case where $\beta_C > 0$, we consider a finite horizon of K discrete time steps, indexed by k. In addition, we simplify the problem setup such that the robotic agent has only two control actions:

- an "informative" action $\alpha_k = 1$, which gains information about the candidate model, so that $\mathbb{E}\left[B_{k+1}(1)\right] \geq B_k(1)$, but has a cost of $c(1, B_k) = 1$ for all B_k; and
- a "null" action $\alpha_k = 0$, which does not gain information about the candidate model, so that $B_{k+1} = B_k$, but has no cost, $c(0, B_k) = 0$ for all B_k.

In addition to the results for the general case, the following lemmas can be proven for this simplified two-action system, which retains the main structure of the general case and, for policy construction, provides an exploitation/exploration trade-off.

Lemma 2. Parametrized family of solutions. *Consider any two policies $\tilde{\bar{\alpha}}$ and $\hat{\bar{\alpha}}$ that respectively generate sequences of actions $\tilde{\alpha}_0\tilde{\alpha}_1 \ldots \tilde{\alpha}_{M-1}$ and $\hat{\alpha}_0\hat{\alpha}_1 \ldots \hat{\alpha}_{M-1}$ for an M step episode with $\sum_{k=0}^{K} \tilde{\alpha}_k = \sum_{k=0}^{K} \hat{\alpha}_k = N$ for any $N \leq M$.*

The cumulative expected entropy reward under both policies is equal:

$$\mathbb{E}\left[\sum_{k=0}^{K} R_k^H(\tilde{B}_k, \tilde{\alpha}_k, \tilde{B}_{k+1})\right] = \mathbb{E}\left[\sum_{k=0}^{K} R_k^H(\hat{B}_k, \hat{\alpha}_k, \hat{B}_{k+1})\right] \tag{15}$$

Lemma 3. Step-function policies. *Consider the "step function" policy $\tilde{\bar{\alpha}}$ with N informative actions, $\tilde{\alpha}_k = 1$ for all $k = 0, \ldots, N-1$, and $\tilde{\alpha}_k = 0$ for all $k = N, \ldots, M$. Any other policy $\hat{\bar{\alpha}}$ with N informative actions that has $\hat{\alpha}_k = 0$ for at least one $k = 0, \ldots, N-1$ has lower cumulative expected KL reward,*

$$\mathbb{E}\left[\sum_{k=0}^{M} R^{KL}(\hat{B}_k, \hat{\alpha}_k, \hat{B}_{k+1})\right] \leq \mathbb{E}\left[\sum_{k=0}^{M} R^{KL}(\tilde{B}_k, \tilde{\alpha}_k, \tilde{B}_{k+1})\right]. \tag{16}$$

Lemma 2 allows us to parametrize policies using the total number of informative actions $\alpha_k = 1$, and Lemma 3 states that for any policy with N informative actions, choosing the first N actions to be informative, $\alpha_k = 1, k = 0, \ldots, N-1$ results in the lowest KL cumulative cost. Since the entropy reward structure is a proxy for the KL reward structure, we can assume that for a policy with N informative actions, $\alpha_k = 1, k = 0, \ldots, N-1$. This observation drastically reduces the policy search space.

Finally, in settings where simulation is used, the KL reward structure is in particular amenable to use in reinforcement learning (RL) algorithms. In other words, the environment simulator can provide the agent feedback in a reward signal that is proportional to deviation of its belief from the ground-truth mode of the other agent. The numerical experiments of Section 6 demonstrate this application in RL.

5 Example Scenarios

In this section, we show how formal methods can capture high-level agent intent in diverse examples ranging from cooperative to adversarial, such as a high-speed car

chase (subsection 5.1) or cooperative lane-merging. In particular, we emphasize a *data-driven* approach to either mine parameters of temporal logic formulae from the Stanford Drone Dataset (subsections 5.2 - 5.4) or learn behaviors using generative models for highway lane-merging (subsection 5.5). Collectively, the selected examples show the generality of our approach in various settings and how temporal logic allows us to filter high-dimensional observations into concise behavioral summaries for tractable decision making.

5.1 Robotic Car-following BLTL Specifications

Our first example is adversarial, where a robot transporting medical supplies must make subtle, costly route deviations to discern the nature of followers (Figure 1(b)). The robot's lane occupancy in one of X total lanes is denoted by system variables $\mathcal{Y} = \{C_1, \ldots, C_X\}$. Likewise, the follower's lane is denoted by environment variables $\mathcal{X} = \{F_1, \ldots, F_X\}$, leading to overall state $s_t = [C_{x,t} F_{x,t}]$ at time t. Follower models are referred to as `pursuant`, $\texttt{surveil}_z$ (for a surveillance car), or `benign` for a civilian. The $\texttt{surveil}_z$ 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 Chinchali et al. (2016) by adding control costs and more surveillance cars:

$$\varphi_{\texttt{benign}} = \texttt{True} \tag{17}$$

$$\varphi_{\texttt{surveil}_z} = \bigwedge_{x \in \{1,\ldots X\}} \Box \left(C_x \implies \Diamond_{[0,T_z]} F_{x-z} \vee F_{x-z+1} \vee \cdots \vee F_{x+z} \right) \tag{18}$$

$$\varphi_{\texttt{pursuant}} = \bigwedge_{x \in \{1,\ldots,X\}} \Box \left(C_x \implies \Diamond_{[0,T_{\texttt{pursuant}}]} F_x \right) \tag{19}$$

That civilian cars have no temporal logic constraint is represented by (17). Notice that $T_{\texttt{pursuant}}$ provides a time bound in (19).

The robot can probe the follower by changing lanes which incurs a lane-deviation and fuel cost of $c(\alpha_k) = 1$. 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 (18) and (19). In large-scale problems with many lanes and long BLTL time bounds T, Figure 2 shows how dimensionality reduction applies in car-following using bitvector observations.

5.2 Stanford Drone Dataset (SDD)

The second example is cooperative and shows how data mining can be used to learn parameters of temporal logic formulae, such as BLTL time bounds. The SDD Robicquet et al. (2016) 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. Though the dataset features entirely human-human interactions of vehicles and pedestrians, we use it to mine specification parameters on safe interaction between agents to guide future autonomous vehicle design.

Specifically, we consider how low-speed robotic carts that transport supplies across campus 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 merging strategies that are richer than avoiding accidents at all cost.

We performed extensive quality-control of over 69 GB of SDD data to identify high-quality motorized trajectories, derive velocities, and compute intra-agent distance distributions. SDD trajectories show how both pedestrians and carts respond to rapidly changing traffic densities and intra-agent distances to merge or cross at opportune times (Figure 4). We mine such behavior from the SDD and incorporate symbolic specifications into a scenario where a robotic cart can *proactively* signal its merging intent to nearby pedestrians. Such behavior is encoded in the following specifications.

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 4(b). Pedestrian models are denoted by $m \in \{\textit{cautious}, \textit{daring}\}$, which exhibit varied propensities to cross based on traffic and robot signals. 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 $\alpha \in \tilde{A}$, a pedestrian of model m will not cross for a time $T^m_{\tilde{\rho},\alpha}$ to safely

assess their surroundings, and after may cross based on their internal risk profile. Such behavior is captured by formula $\varphi^m_{\tilde{\rho},\alpha}$ and an example for a *daring* pedestrian during heavy traffic ($\rho > \rho_0$) after the robot indicates *safe* is:

$$\varphi^{daring}_{h,s} = \Box[\underbrace{(\rho > \rho_0)}_{\text{heavy}} \wedge \underbrace{safe}_{\text{impulse}}] \implies \underbrace{\neg\, cross\ \mathbf{U}_{[0,T^{daring}_{h,s}]}\ \texttt{True}}_{\text{can only cross after wait-time}} \tag{20}$$

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

$$\varphi^m = \bigwedge_{\alpha\in\tilde{A}} \bigwedge_{\tilde{\rho}\in\{h,l\}} \varphi^m_{\tilde{\rho},\alpha} \tag{21}$$

Crucially, BLTL time-bounds $T^m_{\tilde{\rho},\alpha}$ 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 $T(\varphi) > T^m_{\tilde{\rho},\alpha}$ for any model m, traffic condition $\tilde{\rho}$, and safety signal α to assess the pedestrian's response. Suppose the agent indicated *safe* during 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 $T^{daring}_{h,s} << T(\varphi)$, the probability of observing $\neg\, cross$ in a long interval might indicate a *cautious* model.

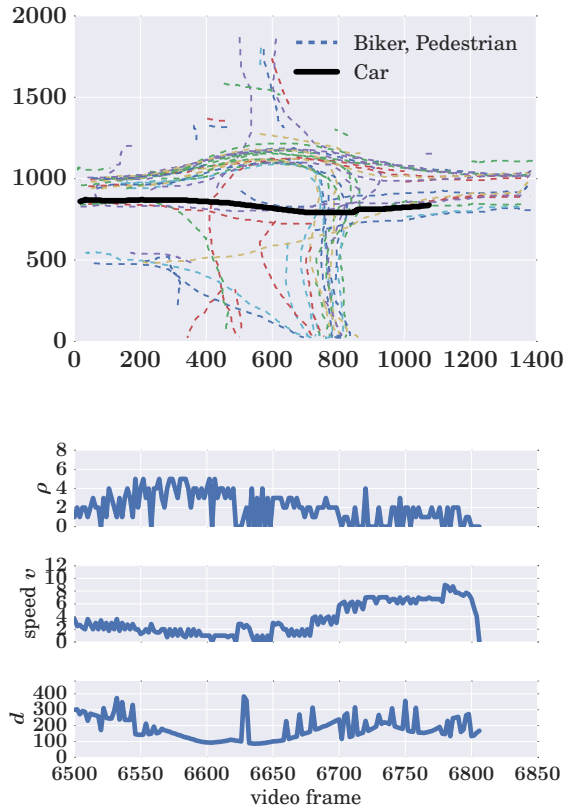

Figure 4. (a, top) Aerial view of an SDD scene where a car (bold) rapidly accelerates to merge into a roundabout. (b, bottom) The car rapidly accelerates, reflected in speed v, when traffic ρ subsides after video frame 6700. Video frames are used as a proxy for continuous time, which is not reported.

5.4 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 *. We can learn such parameters automatically by contrasting velocity and traffic distributions during merging and steady-driving scenarios to find separating thresholds. Formulae (22) and (23) capture scenarios like Figure 4, where a cart cannot signal *safe* as it attempts to merge:

$$\varphi^{slow} = \Box[\underbrace{(\rho > \rho_0)}_{\text{cross-traffic}} \wedge \underbrace{(d < d_0)}_{\text{biker close}} \wedge \underbrace{(v = v_0)}_{\text{speed}}] \implies \Diamond_{[0,T^{slow}]} \underbrace{\left(v \leq \frac{v_0}{M}\right)}_{\text{decelerate}} \tag{22}$$

$$\varphi^{merge} = \Box[\underbrace{(\rho < \rho_0)}_{\text{light traffic}} \wedge \underbrace{(d > d_0)}_{\text{biker far}} \wedge (v = v_0)] \implies \Diamond_{[0,T^{merge}]}[\underbrace{(v \geq Lv_0)}_{\text{accelerate}} \wedge \underbrace{\neg\, safe}_{\text{disallow cross}}] \tag{23}$$

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 $c(\alpha, B_k) \propto H(B_k)$ for any action α.

5.5 Highway Lane Merging Example

The third example is cooperative, and illustrates how distributions of human responses to informative robot probes can be directly learned from data using generative models, such as Conditional Variational Autoencoders (Sohn et al. (2015)). Details of the data-driven approach are provided later in Section 6.2.1.

Consider a highway on-ramp lane merge scenario, depicted in Figure 1(c). One of the cars, denoted R and illustrated in red, is the autonomous system we seek to control in order to complete the lane changing task in a safe,

*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 units of these parameters.

smooth manner. The other car, denoted by H, is human-driven, with an adversarial MDP model $\mathcal{M}_1$ indicating its driving style which needs to be inferred by the robot. We only assume that the human-driven car is described by one of the adversarial MDPs in $\{\mathcal{M}_1, \mathcal{M}_2, \ldots\}$.

Let x_H^t, x_R^t and ℓ_H^t, ℓ_R^t denote the longitudinal and lateral positions of the vehicles along the length and across the lanes of the highway at time t, respectively. Suppose that initially $x_H^0 = x_R^0 = 0$ and $\ell_H^0 = 0.5, \ell_R^0 = -0.5$. Within some time limit T_f and distance d_{end}, the robot needs to switch lanes with the human, requiring the human to slow down and yield for the robot or the robot to speed up and overtake the human. The robot and human can only swap lanes when they are far enough, specifically when the difference of their longitudinal positions is above a safety threshold d_{safety}, expressed by $|x_H^t - x_R^t| > d_{\text{safety}}$.

Let the control cost for an interaction sequence $J_{\text{total}}(\mathcal{H}_T)$ (eq. 24) be the sum of the human and robot costs $J_H(\mathcal{H}_T)$ (eq. 25) and $J_R(\mathcal{H}_T)$ (26). The costs penalize rash, last-minute accelerations to switch lanes, where we normalize accelerations by the maximum limit allowed of $\ddot{x}_{\text{limit}}$.

$$J_{\text{total}}(\mathcal{H}_T) = J_H(\mathcal{H}_T) + J_R(\mathcal{H}_T) \tag{24}$$

$$J_H(\mathcal{H}_T) = \frac{1}{T+1} \sum_{i=0}^{T} \frac{|\ddot{x}_H^t|}{|\ddot{x}_{\text{limit}}|} \tag{25}$$

The robot cost (eq. 26) not only penalizes accelerations, but it also incorporates a lane-change incentive $J_{R,\text{swap}}(\mathcal{H}_T)$ to penalize not swapping lanes especially as the cars approach the road end d_{end} and are not at least d_{safety} distance apart.

$$J_R(\mathcal{H}_T) = J_{R,\text{acc}}(\mathcal{H}_T) + J_{R,\text{swap}}(\mathcal{H}_T) \tag{26}$$

$$J_{R,\text{acc}}(\mathcal{H}_T) = \frac{1}{T+1} \sum_{i=0}^{T} \frac{|\ddot{x}_R^t|}{|\ddot{x}_{\text{limit}}|} \tag{27}$$

The lane-change cost only occurs if the final position difference at time T (at the end of interaction history $\mathcal{H}_T$) $\Delta x^T = |x_H^T - x_R^T|$ is above the safety threshold and depends on whether the further vehicle position is before or after the road-end d_{end}, where the furthest position is indicated by $x_{\max}^T = \max\{x_H^T, x_R^T\}$. In the first case when either car has surpassed the road-end, not meeting the safety distance carries a large magnitude penalty given by J_{crash}. In the second case when both cars still have not reached the end, we modulate by an "urgency" term indicating how much space remains before the road end.

$$J_{R,\text{swap}}(\mathcal{H}_T) = \begin{cases} J_{\text{crash}} \times \mathbb{1}(\Delta x^T < d_{\text{safety}}) & x_{\max}^T \geq d_{\text{end}} \\ \left(1 - \frac{|d_{\text{end}} - x_{\max}^T|}{d_{\text{end}}}\right) \mathbb{1}(\Delta x^t < d_{\text{safety}}) & x_{\max}^T < d_{\text{end}} \end{cases}$$

In order for the autonomous system to complete its goal of swapping lanes with the human, it must predict the intent of the human driver. For example, a driver in a hurry may prefer not to let the other car pass in front; on the other hand a passive driver may prefer to stay back and wait for the other car to make its move. Combining the human's a priori unknown goal and unknown preference, we assume the following BLTL formulas corresponding to the adversarial MDPs:

$$\begin{aligned} H_{\text{switch}} &:= \Diamond_{[0,T_f]}(\ell_H < -0.25) \\ &\quad \wedge \Box(x_H \geq d_{\text{end}} \implies \ell_H < -0.25), \\ &\quad \wedge \Box\left((\ell_H < -0.25) \implies (|x_H - x_R| > d_{\text{safety}})\right), \\ H_{\text{hurry-}\kappa} &:= \Box\left(\Diamond_{[0,T]}(x_H \geq \kappa x_R) \vee |\ell_H - \ell_R| \geq 0.5\right), \end{aligned} \tag{28}$$

$$H_{\text{passive-}\kappa} := \Box\left(\Diamond_{[0,T]}(x_H \leq \kappa x_R) \vee |\ell_H - \ell_R| \geq 0.5\right). \tag{29}$$

As an example, Equation (28) formalizes the goal of "change lanes" for a human driver that has the preference of "being in a hurry". In other words, the human position x_H must eventually be at least κ times the robot position x_R within the BLTL time bound T in each decision making iteration, corresponding to a finite time T of joint interaction along the road. The lane-swap occurs over the duration of several decision making iterations, each of BLTL time bound T. Multiplicative factor κ is henceforth referred to as the *risk factor* since aggressive drivers might have $\kappa >> 1$ indicating they might want to stay quite ahead of the robot position. In subsequent simulations, various degrees of aggressive driving by human drivers in a hurry are referred to by BLTL models $\text{hurry}-\kappa$.

Conversely, Equation (29) formalizes the goal of "change lanes" for a human driver that has the preference of "passively yielding to the robot". In other words, the human position x_H must eventually be at least κ times *below* the robot position x_R within the BLTL time bound T. For passive driving risk factor $\kappa < 1$, indicating the human wants to stay behind the robot position. In subsequent simulations, various degrees of cautious driving by passive human drivers are referred to by BLTL models $\text{passive}-\kappa$.

6 Simulation Results

The principal aim of our evaluation is to show how dimensionality reduction can be used to solve otherwise intractable adversarial MDPs using belief space planning with increasingly sophisticated approaches. Starting from initial examples that can be solved with tree-based value iteration (subsection 6.1), we then address larger problems with several competing models or cases where the ground truth model is not in the candidate set. For these larger examples, we employ data-driven methods such as generative modeling (6.2) or RL with deep neural networks 6.3. Source code that can be used to replicate numerical examples of this section is available at https://github.com/StanfordASL/idwithtasks.

6.1 Value Iteration Tree Results

Figure 5 shows a proactive car-merging strategy solved by BLTL tree value iteration. As introduced in Section 5.4, control costs capture model uncertainty and risk probabilities, so $c(\alpha, B_k) = \frac{c_0(\alpha)}{2}(1 + |\frac{H(B_k)}{H(B_0)}|)$ where $c_o(\text{safe}) = 0.40$, $c_0(\text{unsafe}) = 0.30$, and $c_0(\text{no signal}) = 0$ †. 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 5(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 $T(\varphi)$.

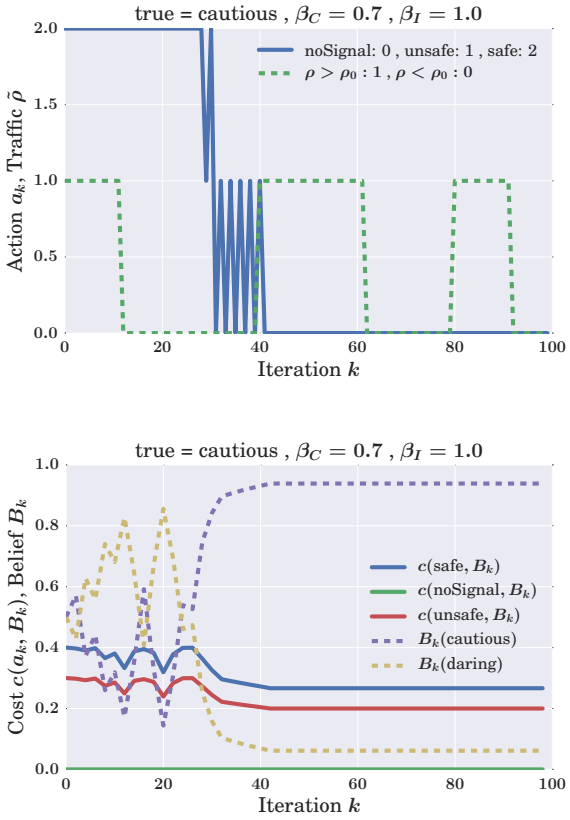

Figure 5. (a, top) The cart initially chooses the high-cost, but most informative $safe$ action when traffic ρ is heavy. As the traffic subsides, the cart chooses the lower cost, but still informative $unsafe$ signal since it can now merge. (b, bottom) Belief B_k indicates we correctly identify the true cautious pedestrian and informative control signals decay in cost over time as we increase model confidence. Eventually, the agent chooses the zero cost $no\ signal$ action once it has high model confidence and has merged.

6.2 Generative Modeling of Human Behavior

6.2.1 Data modeling approach

In large scale problems, an AS must anticipate a wide variety of human reponses to potential robot control sequences in order to plan low-cost, informative probes. Often, such a probability distribution over high-dimensional human future responses to candidate robot control vectors cannot be analytically modeled, requiring one to *learn* such distributions from data. In particular, Schmerling et al. (2018) shows promising results for a highway on-ramp lane merging scenario where distributions of human acceleration profiles conditioned on candidate robot controls (accelerations) are learned using a deep-neural network based generative model, Conditional Variational Autoencoders (CVAEs), as demonstrated in prior work by Sohn et al. (2015). (Also, consult the tutorial on VAEs by Doersch (2016).)

We did not have access to a real lane-merging dataset with recoverable temporal logic formulae codifying diverse

†This is just one representative example of control costs allowed by our general framework.

driver styles. Hence, we focused on a highway lane-merging example where human behavior data is *synthetically* generated according to agent-intents encoded in BLTL. Although such generative models would need to be learned from data in a real engineering use-case, the principal aim of our study is to illustrate how even a high-dimensional synthetic dataset can be used to inform a tractable control algorithm using the lens of temporal logic.

6.2.2 Application to highway merging

As introduced in Section 5.5, robot and human control vectors are accelerations along a lane, and a lane-swap can only occur if the position of both vehicles is beyond a minimum safety distance. In our simulations, initial lane positions of the robot and human are given by $x_R^0 = 5$ and $x_H^0 = 0$ meters and both start at the same velocity of $\dot{x}_R^0 = 10\ m/s^2$ and $\dot{x}_H^0 = 10\ m/s^2$. Each decision making iteration for the belief space planner k lasts a BLTL time bound of $T = 6s$ of interaction between the robot and human along the road. Every window of 2 seconds, the robot and human can choose accelerations along the lane of $\ddot{x}_{H,R}^t \in [0, 1, 3, -1, -3]\ m/s^2$. Hence, a probe α_k for the robot is a planned acceleration vector for $T = 6s$ of 3 choices from the discretized acceleration set, such as $\alpha_k = [1, -1, -3]\ m/s^2$, since an acceleration is applied on a window of $2s$. For simplicity of simulation, we do not explicitly solve for lateral accelerations of the cars in the ℓ dimension. Rather, we make the natural assumption that when the lane positions x are sufficiently far apart past a safety distance d_{safety}, the robot and human can safely swap lanes, codified by $|x_H^t - x_R^t| > d_{\text{safety}}$.

If the robot and human are close in position and the robot accelerates, aggressive human models of the form $\text{hurry}{-}\kappa$ must accelerate considerably to eventually remain ahead of the robot in the BLTL time bound T. In particular, if the risk factor κ increases, the set of feasible human responses to a robot acceleration become increasingly constrained and more aggressive to adhere to the specification.

We created generative models that provide a probability distribution over feasible human accelerations conditioned on a robot candidate acceleration and a BLTL model such as $\text{hurry}{-}\kappa$. Given a future robot acceleration and initial positions and velocities for the robot and human, we use simple integrator dynamics to project the robot's future position. Searching over a discretized set of human acceleration vectors, we return a probability distribution of plausible human accelerations which yield joint human/robot positions that adhere to the BLTL formula of interest such as $\text{hurry}{-}\kappa$. The probability of feasible human accelerations is weighted by the control cost to encode the notion that low-cost, *feasible* human responses are more likely.

Figures 6(a) and 6(b) illustrate our generative model, where samples from a probability distribution of human accelerations (red) are shown in response to a planned constant-acceleration robot control (dashed-blue) where the robot and human start close together and at the same velocity. As expected, a more stringent BLTL model of $\text{hurry}{-}1.35$ (Fig. 6(a)), where the human must stay ahead of the robot by a larger margin than the $\text{hurry}{-}1.20$ model (Fig. 6(b)), leads to fewer, but more aggressive feasible human responses.

6.2.3 Belief-space simulation leveraging generative model

We conducted a simulation for the highway lane-merging scenario introduced in Section 5.5, where a robot must choose low-cost probes (acceleration vectors) to determine if a human driver is passive or aggressive in order to coordinate a lane-change. The robot has access to a generative model for various human driver styles, such as in Figure 6(a), which produces high-dimensional human response distributions for planned robot probes over a BLTL time duration T. Since accounting for such a continuum of responses from generative models is intractable, the robot clusters responses into concise observation bitvectors describing distinct driver styles using the Belief MDP and tree search approach introduced in Problem 2 and Figure 3.

As illustrated in Figure 7(a), the set of plausible candidate models the robot assumes may occur, and therefore must differentiate, includes three $\text{hurry}{-}\kappa$ models where risk factor $\kappa \in [1.05, 1.09, 1.20]$ and two markedly different $\text{passive}{-}\kappa$ models where risk factor $\kappa \in [0.90, 0.70]$.

Notably, the true human model of $\text{hurry}{-}1.10$ is *not* in the robot's set of plausible candidate models, so we test whether it can identify the "best-fit" model of $\text{hurry}{-}1.09$. Such a situation might often occur in practice when a robot's assumptions differ from operating conditions.

Figure 7 illustrates the results of a successful lane-swap where the robot's high-level belief over human driver styles is shown in Figure 7(a) and its low-level state and acceleration controls are given in Figure 7(b). In this successful run, the robot is overtaken by a human driver of BLTL model $\text{hurry}{-}1.10$, who hurries ahead and creates ample distance for both agents to change lanes (Fig. 7(b), position plot). Initially, at iteration $k = 0$, the robot believes all models to be equiprobable (belief plot, Fig. 7(a)) and chooses to accelerate until about $t = 2$ (Fig. 7(b), acceleration plot). However, as the aggressive human driver accelerates in response (red), the robot starts to realize a

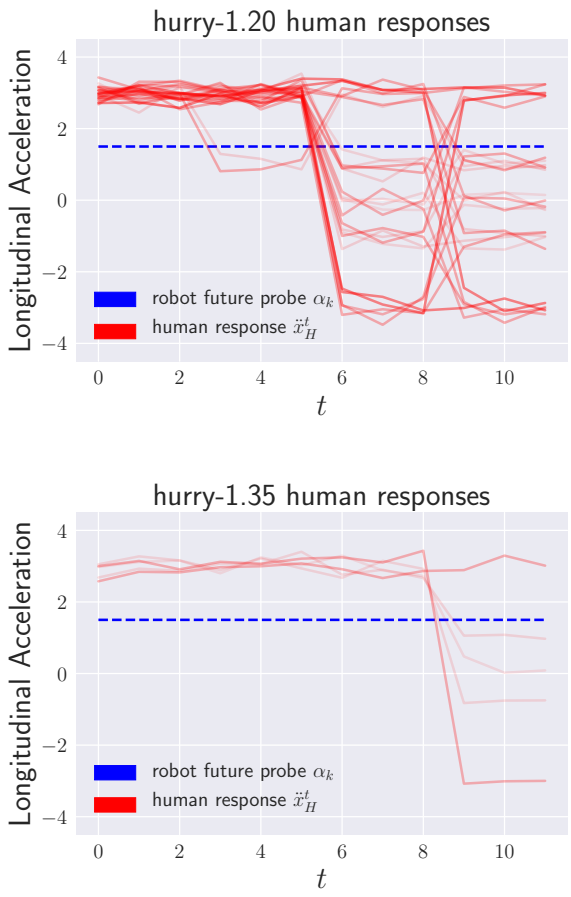

Figure 6. Generative model of human responses for various BLTL formulae describing driver styles.

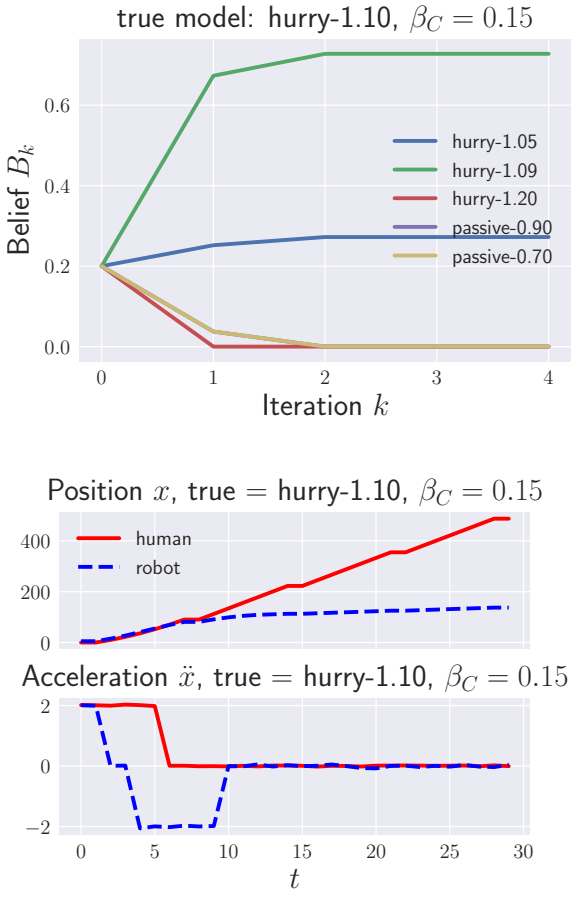

Figure 7. (a, top) Belief distribution centers on closest human model to the ground truth. (b, bottom) The human in a hurry overtakes the robot to create ample distance for a lane-swap.

hurry model is more likely, after which it decelerates from $t = 2$ to $t = 10$ to yield to the human and allow sufficient room for a safe lane-change, after which it keeps a constant velocity to minimize acceleration cost.

As desired, Figure 7(a) illustrates the robot places highest belief in the "best-fit" model hurry-1.09 (green) which is virtually indistinguishable from the true model of hurry-1.10. The robot easily realizes that the human driver is not passive nor too aggressive such as hurry-1.20. Naturally, it places lower weight on a similar model of hurry-1.05 (blue) since many trajectories that match this model also match those of the true model, but not as closely as the "best-fit" of hurry-1.09.

In essence, clustering high-dimensional interaction histories from a generative model into observation bitvectors allows for tractable planning in belief space, allowing simultaneous identification of true human driver style and a safe lane-interchange in the highway example.

6.3 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 α_k based on current belief state B_k using parametrized control policy $\pi_\theta(B_k)$, where parameters θ are learned over time. The environment generates observations o_k under true model $\mathcal{M}_1$ 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.

A proactive decision making scheme can first be trained with an RL simulator where the environment updates the belief vector for several ground-truth settings, and then the agent can be deployed in practice. We developed simulators for both examples using the *openAI gym* framework Brockman et al. (2016). We used Google's Tensorflow Abadi et al. (2016) to learn a stochastic control policy using the Actor-Critic (AC) RL learning algorithm Konda and

Tsitsiklis (2000), where the policy $\pi_\theta(B_k)$ is encoded in a neural network with parameters θ of 1 hidden layer of 50 units. Our general framework allows for deeper neural networks to be used for problems with more complex human-robot interaction models. Figure 8(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.

6.3.1 RL Reward Structures

We now highlight several interesting properties of our framework in the RL setting by comprehensively evaluating its performance and convergence on different reward functions or control costs. 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 $\bar{B} = [1, 0, \dots 0]$ and the current belief (Equation 10). The KL 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.

Interestingly, for a wide spectrum of weights β_C, β_I, the policy learned under the KL reward converged faster than the entropy reward for the same experimental settings (Figure 8(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 8(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. As often seen in practical RL, many trials of the entropy policy showed it does not learn for several initial episodes as it explores sub-optimal policies, but sees a sudden performance jump as it discovers a better set of policy parameters (Figure 8(b)).

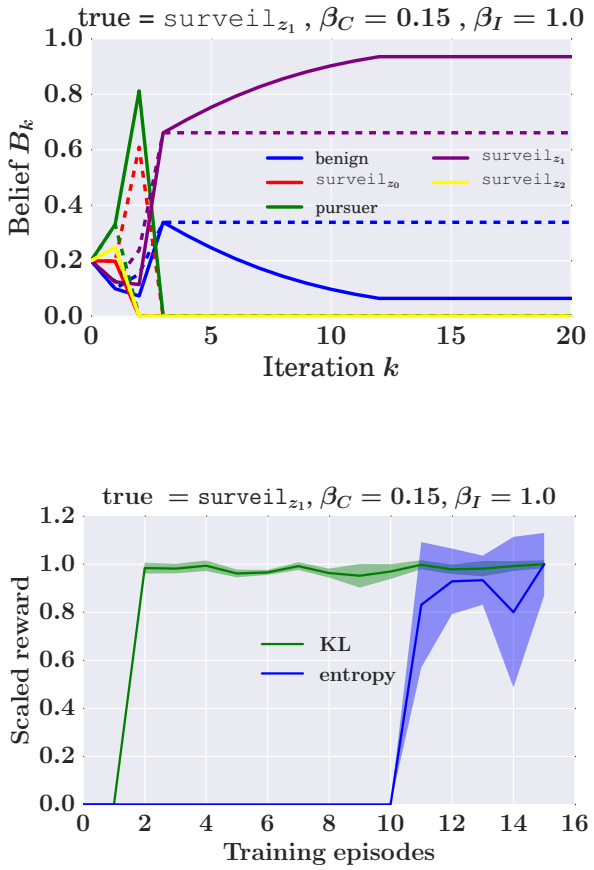

Figure 8. (a, top) In the car-following scenario solved with RL, both KL and entropy rewards lead to correct model identification, but the KL trained policy (solid) identifies true model surveil$_{z_1}$ with higher certainty. (b, bottom) Normalized RL learning curves for both reward functions indicates KL converges faster with lower variance.

6.3.2 Trading off informative probes with control cost

We now show that our RL agent's policies can flexibly trade off exploration of human agent intent, which incurs a control cost weighted by β_C in the reward function, with exploitation of the currently believed model. In particular, as informative probes become costlier by larger weights β_C, the agent should probe less, resulting in less certainty in the true model measured in the belief distribution. As expected, Figure 9 shows that as β_C increases for a car-following example solved with RL, the robot has less certainty but still identifies the true model. The dependence of the converged belief on control cost weight β_C is not purely monotonic

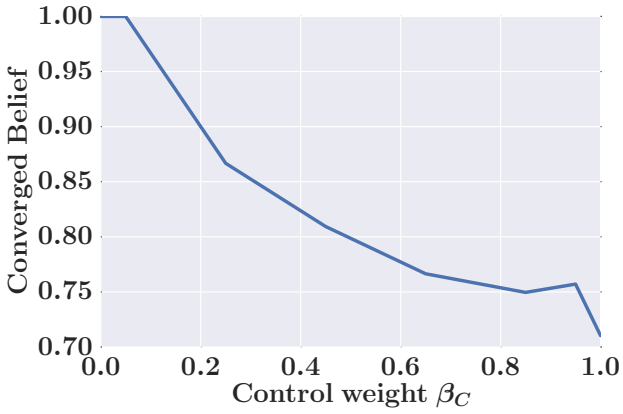

Figure 9. An RL agent adapts its policy to the reward function weight for the control cost β_C. As the control cost penalty β_C increases, the agent uses less informative probes but still identifies the true model, albeit with less certainty in the belief vector.

since the converged belief was the average of several runs from a *stochastic* RL policy.

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 conducting experimental user studies where a simulated autonomous cart signals its merging intent to human pedestrian subjects using a ProDM scheme, allowing us to *explicitly probe* for human risk profiles. Such experiments will likely require us to expand our specification mining approach to automatically derive *both* the specifications and associated parameters from high-dimensional datasets. In a real deployment, we would also need to couple such automatically derived specifications with a safety mechanism that defaults to a conservative AS policy if experimental data deviate significantly from learned models. We also plan to leverage the highway lane-merging dataset generated by real humans and CVAE generative model from Schmerling et al. (2018) for our temporal logic based scheme. In particular, such an analysis would require us to *automatically mine* concise temporal logic formulae from their dataset.

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.

Acknowledgements

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

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y and Zheng X (2016) TensorFlow: A system for large-scale machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association. ISBN 978-1-931971-33-1, pp. 265–283. URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

Baier C and Katoen JP (2008) *Principles of Model Checking*. MIT Press. ISBN 9780262026499. URL https://mitpress.mit.edu/books/principles-model-checking.

Braziunas D (2003) POMDP solution methods. Technical report, Department of Computer Science, University of Toronto. URL https://www.cs.toronto.edu/~darius/papers/POMDP_survey.pdf.

Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J and Zaremba W (2016) OpenAI gym. Available at https://arxiv.org/abs/1606.01540.

Chinchali SP, Livingston SC, Pavone M and Burdick JW (2016) Simultaneous model identification and task satisfaction in the presence of temporal logic constraints. In: *Proc. IEEE Conf. on Robotics and Automation*.

Doersch C (2016) Tutorial on variational autoencoders. Available at: https://arxiv.org/abs/1606.05908.

Gmytrasiewicz PJ and Doshi P (2005) A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24: 49–79.

Javdani S, Srinivasa SS and Bagnell JA (2015) Shared autonomy via hindsight optimization. In: *Robotics: Science and Systems*. DOI:10.15607/RSS.2015.XI.032.

Jones A, Schwager M and Belta C (2015) Information-guided persistent monitoring under temporal logic constraints. In: *American Control Conference*. pp. 1911–1916.

Knight W (2016) New self-driving car tells pedestrians when it's safe to cross the street. *MIT Technology Review* URL https://www.technologyreview.com/s/602267/new-self-driving-car-tells-pedestrians-when-its-safe-to-cross-the-street/.

Konda VR and Tsitsiklis JN (2000) Actor-critic algorithms. In: *Advances in Neural Information Processing Systems*. MIT Press, pp. 1008–1014. URL http://papers.nips.cc/paper/1786-actor-critic-algorithms.pdf.

Koymans R (1990) Specifying real-time properties with metric temporal logic. *Real-Time Systems* 2: 255–299. DOI:10.1007/BF01995674.

Madani O, Hanks S and Condon A (1999) On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In: *Proc. AAAI Conf. on Artificial Intelligence*. URL https://www.aaai.org/Papers/AAAI/1999/AAAI99-077.pdf.

Nguyen THD, Hsu D, Lee WS, Leong TY, Kaelbling LP, Lozano-Perez T and Grant AH (2011) Capir: Collaborative action planning with intention recognition. In: *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*.

Papadimitriou CH and Tsitsiklis JN (1987) The complexity of Markov decision processes. *Mathematics of Operations Research* 12(3): 441–450.

Raman V, Donzé A, Sadigh D, Murray RM and Seshia SA (2015) Reactive synthesis from signal temporal logic specifications. In: *Hybrid Systems: Computation and Control*. pp. 239–248. DOI:10.1145/2728606.2728628.

Robicquet A, Sadeghian A, Alahi A and Savarese S (2016) Learning social etiquette: Human trajectory understanding in crowded scenes. In: *European Control Conference*.

Sadigh D, Sastry SS, Seshia SA and Dragan A (2016) Information gathering actions over human internal state. In: *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*. pp. 66–73. DOI: 10.1109/IROS.2016.7759036.

Schmerling E, Leung K, Vollprecht W and Pavone M (2018) Multimodal probabilistic model-based planning for human-robot interaction. In: *Proc. IEEE Conf. on Robotics and Automation*.

Sohn K, Lee H and Yan X (2015) Learning structured output representation using deep conditional generative models. In: *Advances in Neural Information Processing Systems*. pp. 3483–3491. URL http://papers.nips.cc/paper/5775-learning-structured-output-representation-using-deep-conditional-generative-models.pdf.

Trautman P and Krause A (2010) Unfreezing the robot: Navigation in dense, interacting crowds. In: *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*. pp. 797–803. DOI:10.1109/IROS.2010.5654369.

Wongpiromsarn T and Frazzoli E (2012) Control of probabilistic systems under dynamic, partially known environments with temporal logic specifications. In: *Proc. IEEE Conf. on Decision and Control*. pp. 7644–7651. DOI:10.1109/CDC.2012.6426524.

Appendix

Details of Preliminaries and Formulation

In this section, additional details of the formulation presented in Section 3 are provided for completeness. These are not critical for understanding the main results, but some readers may find them to be useful.

Let $\mathcal{M}$ be a labeled adversarial MDP. The set of interaction histories of duration T and consistent with strategies π and μ, which is denoted by $\mathrm{Hist}(\mathcal{M}, T, \pi, \mu)$, is defined in (2). If the constraining strategies are removed, we obtain the set of interaction histories of duration T that can occur under some sequence of actions,

$$\mathrm{Hist}(\mathcal{M}, T) = \big\{\mathcal{H}_T \mid s_0 \in \mathrm{Init} \wedge \forall t < T : \mathbf{P}(s_t, a_{t,1}, a_{t,2}, s_{t+1}) > 0, \text{ where } a_{t,1} \in \mathrm{Act}^c,\ a_{t,2} \in \mathrm{Act}^u \big\}. \quad (30)$$

Clearly $\mathrm{Hist}(\mathcal{M}, T, \pi, \mu) \subseteq \mathrm{Hist}(\mathcal{M}, T)$.

Construction, existence, and uniqueness of the label function $\mathcal{L}$ that is introduced in Section 3.3 are shown here.

The dependence of each $\sigma \in \mathrm{Traces}(\mathcal{M}, T, \pi, \mu)$ in (3) on some $\mathcal{H}_T \in \mathrm{Hist}(\mathcal{M}, T, \pi, \mu)$ can be generalized to show that for each $\mathcal{H}_T \in \mathrm{Hist}(\mathcal{M}, T, \pi, \mu)$, there is a unique $\sigma \in \mathrm{Traces}(\mathcal{M}, T, \pi, \mu)$ associated with it. Therefore, we can define a function $\mathcal{L}$ from $\mathrm{Hist}(\mathcal{M}, T, \pi, \mu)$ onto $\mathrm{Traces}(\mathcal{M}, T, \pi, \mu)$ consistent with the comprehension in (3), i.e., such that for all $\mathcal{H}_T \in \mathrm{Hist}(\mathcal{M}, T, \pi, \mu)$, $\mathcal{L}(\mathcal{H}_T) \in \mathrm{Traces}(\mathcal{M}, T, \pi, \mu)$ and $\mathcal{H}_T$ realizes the existential quantification in the definition of Traces.

The sketch of a proof for the existence and uniqueness of $\mathcal{L}$ is as follows. Let $\mathcal{H}_T \in \mathrm{Hist}(\mathcal{M}, T, \pi, \mu)$. Observe from the definition of interaction history that $\mathcal{H}_T$ defines a finite sequence of $T+1$ states, call it $\mathbf{S}(\mathcal{H}_T)$, which is obtained by removing the actions in $\mathcal{H}_T$. The labeling function L (from the labelled adversarial MDP $\mathcal{M}$) is defined on the domain $\mathbf{S}$, so applying L to each item in the finite sequence $\mathbf{S}(\mathcal{H}_T)$ yields a finite sequence of $T+1$ elements of $2^{\Pi} = \Sigma$. It is immediate from this construction that this new sequence is an element of $\mathrm{Traces}(\mathcal{M}, T, \pi, \mu)$.

7.1 Proof of Lemma 1

Forward direction: Using $\bar{\alpha}$ implies $\lim_{k\to\infty} B_k(1) = 1$.

First, recall the belief update function given in (8), rewritten here for convenience:

$$B_{k+1}(i) = \eta B_k(i) \Pr(o_k \mid \alpha_k, \mathcal{M}_i)$$

Consider the special case when $\Pr(o \mid \alpha, \mathcal{M}_j) = 0$ for some j, while $\Pr(o \mid \alpha, \mathcal{M}_1) > 0$. If this action α is infinitely often selected by $\bar{\bar{\alpha}}$, then with probability 1, α will be chosen at some time $\hat{k}$, and we would have $B_{\bar{k}}(j) = 0$ at some finite $\bar{k} = \hat{k} + 1$. Furthermore, by (8), we also have $B_k(j) = 0$ for all $k \geq \bar{k}$. Because this occurs independently of the belief vector values and eventually occurs with probability 1, it suffices to consider model indices j for which this property does not hold.

Next, we proceed with some definitions. For $j \neq 1$, define the following functions into sets of satisfaction observation bitvectors:

$$U_j(a) = \{o \mid \Pr(o \mid \alpha, \mathcal{M}_1) > \Pr(o \mid \alpha, \mathcal{M}_j)\} \quad (31)$$

$$L_j(a) = \{o \mid \Pr(o \mid \alpha, \mathcal{M}_1) < \Pr(o \mid \alpha, \mathcal{M}_j)\} \quad (32)$$

where α is any abstract action. By (14), for every j there is some α such that $|U_j(\alpha)| > 0$ and $|L_j(\alpha)| > 0$. Now, for an infinite sequence of satisfaction observation bitvectors $o_0 o_1 o_2 \cdots$ and an infinite sequence of abstract actions $\alpha_0 \alpha_1 \alpha_2 \cdots$, define the sequence of counting maps

$$\Theta_k(U_j) = \left|\left\{\hat{k} \mid 0 \leq \hat{k} \leq k \wedge o_{\hat{k}} \in U_j(\alpha_{\hat{k}})\right\}\right| \quad (33)$$

$$\Theta_k(L_j) = \left|\left\{\hat{k} \mid 0 \leq \hat{k} \leq k \wedge o_{\hat{k}} \in L_j(\alpha_{\hat{k}})\right\}\right| \quad (34)$$

for $k \geq 0$. Note that the sequences of values are nondecreasing, i.e., $0 \leq \Theta_k(U_j) \leq \Theta_{t+1}(U_j)$ and $0 \leq \Theta_k(L_j) \leq \Theta_{t+1}(L_j)$.

Now, we will complete the proof with the following two steps:

1. Show that under $\bar{\bar{\alpha}}$, we have for all $j, \lim_{k\to\infty}(\Theta_k(U_j) - \Theta_k(L_j)) = \infty$.
2. Show that $\lim_{k\to\infty}(\Theta_k(U_j) - \Theta_k(L_j)) = \infty$ implies $\Pr(\lim_{k\to\infty} B_k(1)) = 1$.

For the first step, suppose the contrary, then there is some j such that with positive probability,

$$\exists m, \forall K, \exists k \geq K : \Theta_k(U_j) - \Theta_k(L_j) \leq m. \quad (35)$$

Let o and α be a pair that satisfies (14) and such that the abstract action α is infinitely often selected by $\bar{\bar{\alpha}}$. an estimate of the distribution of $\Pr(o \mid \alpha)$ can be obtained through ζ_k, defined as follows:

$$\zeta_k = \begin{cases} \frac{\mathrm{Count}_k(o,\alpha)}{\mathrm{CountAct}_k(\alpha)} & \text{if } \mathrm{CountAct}_k(\alpha) \neq 0 \\ 0 & \text{otherwise,} \end{cases} \quad (36)$$

where $\mathrm{Count}_k(o, \alpha)$ is the number of occurrences of o given α in the finite sequences $o_0 o_1 \cdots o_k$ and $\alpha_0 \alpha_1 \cdots \alpha_k$, respectively, and $\mathrm{CountAct}_k(\alpha)$ is the number of occurrences of α in the finite sequence $\alpha_0 \alpha_1 \cdots \alpha_k$. Notice that by construction of $\bar{\bar{\alpha}}$, $\mathrm{CountAct}_k(\alpha) \neq 0$ for some k with probability 1.

From the Law of Large Numbers, we have $\lim_{k\to\infty} \zeta_k = \Pr(o \mid \alpha)$. However, the hypothesis (35) implies that for the given o and α, the ratio of the number of times that $\Pr(o \mid \alpha, \mathcal{M}_1) > \Pr(o \mid \alpha, \mathcal{M}_j)$ occurs to the number of times $\Pr(o \mid \alpha, \mathcal{M}_1) < \Pr(o \mid \alpha, \mathcal{M}_j)$ approaches equality. Therefore, we must have $\Pr(o \mid \alpha, \mathcal{M}_1) = \Pr(o \mid \alpha, \mathcal{M}_j)$, which contradicts the assumption in the statement of the Lemma. This proves the first step.

For the second step, suppose that with the policy $\bar{\bar{\alpha}}$, $\lim_{k\to\infty}(\Theta_k(U_j) - \Theta_k(L_j)) = \infty$ with probability 1. Recalling the belief update function in (8), also at the beginning of this proof, and expanding the expression for a sequence of satisfaction observation bitvectors $o_0 o_1 o_2 \cdots$ and abstract actions $\alpha_0 \alpha_1 \alpha_2 \cdots$, we have

$$\begin{aligned} B_k(i) &= \eta_{k-1} B_{k-1}(i) \Pr(o_{k-1} \mid \alpha_{k-1}, \mathcal{M}_i) \\ &= B_{k-2}(i) \cdot \eta_{k-2}\eta_{k-1} \cdot \\ &\qquad \Pr(o_{k-2} \mid \alpha_{k-2}, \mathcal{M}_i) \Pr(o_{k-1} \mid \alpha_{k-1}, \mathcal{M}_i) \\ &\;\;\vdots \\ &= B_0(i) \prod_{\hat{k}=0}^{k-1} \eta_{\hat{k}-1} \Pr(o_{\hat{k}} \mid \alpha_{\hat{k}}, \mathcal{M}_i) \end{aligned}$$

where the $\cdot$ symbol denotes multiplication, and $\eta_{\hat{k}}$ is the normalization factor at time $\hat{k}$ such that the sum of elements of $B_{\hat{k}+1}$ is 1.

From the above, we can divide $B_k(i), i \neq 1$ by $B_k(1)$ and obtain

$$\frac{B_k(i)}{B_k(1)} = \prod_{\hat{k}=0}^{k-1} l_{\hat{k}}, \text{ where}$$

$$l_{\hat{k}} = \frac{\Pr(o_{\hat{k}} \mid \alpha_{\hat{k}}, \mathcal{M}_i)}{\Pr(o_{\hat{k}} \mid \alpha_{\hat{k}}, \mathcal{M}_1)}$$

If $\lim_{k\to\infty}(\Theta_k(U_j) - \Theta_k(L_j)) = \infty$, then the number of times that $\Pr(o \mid \alpha, \mathcal{M}_1) > \Pr(o \mid \alpha, \mathcal{M}_j)$ occurs compared to number of times $\Pr(o \mid \alpha, \mathcal{M}_1) < \Pr(o \mid \alpha, \mathcal{M}_j)$ diverges. This implies that as $k \to \infty$, there are infinitely many $\hat{k}$ such that the ratio $l_{\hat{k}}$ is less than 1.

Furthermore, we must have that for all i, for all K, and some M,

$$\Pr(o_k \mid \alpha_k, \mathcal{M}_1) \neq 0 \quad (38a)$$

(otherwise o_k would never occur in response to α_k), (38b)

$$\frac{\Pr(o_{\hat{k}} \mid \alpha_{\hat{k}}, \mathcal{M}_i)}{\Pr(o_{\hat{k}} \mid \alpha_{\hat{k}}, \mathcal{M}_1)} \leq M \quad (38c)$$

(there are finitely many possible values of α_k and o_k). (38d)

Therefore,

$$\lim_{k\to\infty} \frac{B_k(i)}{B_k(1)} = 0,$$

which implies $\Pr\left(\lim_{k\to\infty} B_k(1)\right) = 1$.

Reverse direction: $\Pr\left(\lim_{k\to\infty} B_k(1)\right) = 1$ implies a policy $\bar{\alpha}$ is used.

Suppose there is a policy $\hat{\bar{\alpha}}$ such that for some $j \neq 1$, an abstract action is not infinitely often chosen that satisfies (14). This means that there is some time K such that for all $k \geq K$, $\hat{\bar{\alpha}}$ selects abstract action α such that

$$\Pr(o \mid \alpha, \mathcal{M}_j) = \Pr(o \mid \alpha, \mathcal{M}_1)$$

for all satisfaction observation bitvectors o.

Then, by the definition belief update function in (8), for all $k \geq K$, we have

$$B_{k+1}(j) = \eta B_k(j) \Pr(o_k \mid \alpha_k, \mathcal{M}_j),$$
$$B_{k+1}(1) = \eta B_k(j) \Pr(o_k \mid \alpha_k, \mathcal{M}_1).$$

Dividing the $B_{k+1}(1)$ by $B_{k+1}(j)$, we obtain

$$\frac{B_{k+1}(1)}{B_{k+1}(j)} = \frac{B_k(1)}{B_k(j)} \tag{39}$$

This implies that $\lim_{k\to\infty} B_k(1) \neq 1$, since otherwise $\lim_{k\to\infty} B_k(j) = 0$, and $\frac{B_k(1)}{B_k(j)}$ would diverge, contradicting (39). ■

7.2 Proof of Theorem 1

Because $\gamma = 1$ and $\beta_C = 0$ by hypothesis, the objective function of the Problem 2 is

$$\lim_{k\to\infty} \mathbb{E}\left(\sum_{k=0}^{K} \beta_I \left(H(B_k) - H(B_{k+1})\right)\right).$$

We first simplify the objective function. Since β is positive and constant, it can be moved outside the limit; in addition, the set of optimal policies is the same for any positive constant β_I, so WLOG, let $\beta_I = 1$. Thus,

$$\lim_{k\to\infty} \mathbb{E}\left(\sum_{k=0}^{K} \left(H(B_k) - H(B_{k+1})\right)\right) \tag{40a}$$
$$= \lim_{k\to\infty} \mathbb{E}\left(H(B_0) - H(B_k)\right) \tag{40b}$$
$$= H(B_0) - \lim_{k\to\infty} \mathbb{E}\left(H(B_k)\right), \tag{40c}$$

where the first equality is a result of repeated alternating coefficients of 1 and -1, causing intermediate values to sum to zero. The second equality is due to B_0 being the fixed uniform distribution according to Section 4.2, and therefore constant under expectation. Since B_0 is a constant, we consider the following optimization problem, whose optimal solutions are the same as those of Problem 2 with $K \to \infty$:

$$\max_{\bar{\alpha}} \left\{ -\lim_{k\to\infty} \mathbb{E}\left(H(B_k)\right) \right\} \tag{41}$$

The standard definition of the entropy for the discrete probability mass function that is represented by the belief vector B is

$$H(B) = -\sum_{i=1}^{N} B \log_2 B.$$

The set of minima of H is exactly the set of standard basis vectors of $\mathbb{R}^N$, with $B(j) = 1$ for some j, and $B = 0$ for all $i \neq j$. The set of minima of H is equal to the set of maxima of $-H$.

Given any policy $\bar{\tilde{\alpha}}$ described in Lemma 1, we have $\lim_{k\to\infty} B_k(1) = 1$. So for all $i \neq 1, \lim_{k\to\infty} B_k(i) = 0$, and therefore $\bar{\tilde{\alpha}}$ is the optimal policy when $\beta_C = 0, \gamma = 1$. ■

7.3 Proof of Lemma 2

In the null control case, $\alpha_k = 0$, we have $B_{k+1} = B_k$ with certainty so the stage reward is

$$\mathbb{E}\left[R^H(B_k, \alpha_k = 0, B_{k+1})\right] = -\beta_C c(0) + H(B_k) - H(B_k) = 0. \tag{42}$$

In the informative control case, $\alpha_k = 1$, the stage reward is

$$\mathbb{E}\left[R^H(B_k, \alpha_k = 1, B_{k+1})\right] = -\beta_C + \mathbb{E}\left[H(B_k)\right] - \mathbb{E}\left[H(B_{k+1})\right]. \tag{43}$$

Let $\mathcal{I}$ and $\mathcal{L}$ be the sets of indices for which $\tilde{\alpha}_k = 1$ and $\hat{\alpha}_k = 1$, respectively.

With policy $\tilde{\bar{\alpha}}$, the cumulative reward is

$$\mathbb{E}\left[\sum_{k=0}^{K} R_k^H(\tilde{B}_k, \tilde{\alpha}_k, \tilde{B}_{k+1})\right]$$
$$= \mathbb{E}\left[\sum_{k\in\mathcal{I}} R_k^H(\tilde{B}_k, \tilde{\alpha}_k, \tilde{B}_{k+1})\right] + \mathbb{E}\left[\sum_{k\notin\mathcal{I}} R_k^H(\tilde{B}_k, \tilde{\alpha}_k, \tilde{B}_{k+1})\right] \tag{44a}$$
$$= \mathbb{E}\sum_{k\in\mathcal{I}} \left\{-\beta_C + H(\tilde{B}_k) - H(\tilde{B}_{k+1})\right\} + \sum_{k\notin\mathcal{I}} 0 \tag{44b}$$
$$= -N\beta_C + \mathbb{E}\left[\sum_{k\in\mathcal{I}} \left\{H(\tilde{B}_k) - H(\tilde{B}_{k+1})\right\}\right]. \tag{44c}$$

With policy $\hat{\bar{\alpha}}$, the cumulative reward is

$$\mathbb{E}\left[\sum_{k=0}^{K} R_k^H(\hat{B}_k, \hat{\alpha}_k, \hat{B}_{k+1})\right]$$
$$= \mathbb{E}\left[\sum_{k\in\mathcal{L}} R_k^H(\hat{B}_k, \hat{\alpha}_k, \hat{B}_{k+1})\right] + \mathbb{E}\left[\sum_{k\notin\mathcal{L}} R_k^H(\hat{B}_k, \hat{\alpha}_k, \hat{B}_{k+1})\right] \tag{45a}$$
$$= \mathbb{E}\left[\sum_{k\in\mathcal{L}} \left\{-\beta_C + H(\hat{B}_k) - H(\hat{B}_{k+1})\right\}\right] + \sum_{k\notin\mathcal{L}} 0 \tag{45b}$$
$$= -N\beta_C + \mathbb{E}\left[\sum_{k\in\mathcal{L}} \left\{H(\hat{B}_k) - H(\hat{B}_{k+1})\right\}\right]. \tag{45c}$$

The elements of $\{\tilde{B}_k\}_{k\in\mathcal{I}}$ and $\{\hat{B}_k\}_{k\in\mathcal{L}}$ are equal, since $B_{k+1} = B_k$ when $\alpha_k = 0$. Thus, ignoring identical belief states B_k caused by the null actions $\alpha_k = 0$, the two policies $\tilde{\bar{\alpha}}$ and $\hat{\bar{\alpha}}$ produce the same sequence of unique belief states in expectation. Thus, the two sums $\sum_{k\in\mathcal{I}}\{H(B_k) - H(B_{k+1})\}$ and $\sum_{k\in\mathcal{L}}\{H(B_k) - H(B_{k+1})\}$ are equal in expectation. ■

7.4 Proof of Lemma 3

Given any B_k, we have that

$$\begin{aligned}\mathrm{KL}(B_k, \bar{B}) &= \sum_i \bar{B}(i) \log \frac{\bar{B}(i)}{B_k(i)} \\ &= -\log B_k(1)\end{aligned}$$

For the $\tilde{\bar{\alpha}}$ with N informative actions, we have

$$\mathbb{E}\left[\sum_{k=0}^{M} R^{KL}(\tilde{B}_k, \tilde{\alpha}_k, \tilde{B}_{k+1})\right] \tag{46a}$$
$$= -N\beta_C + \mathbb{E}\left[\sum_{k=0}^{M} \log \tilde{B}_k(1)\right]$$
$$= -N\beta_C + \mathbb{E}\left[\sum_{k=0}^{N-1} \log \tilde{B}_k(1)\right] + \mathbb{E}\left[\sum_{k=N}^{M} \log \tilde{B}_k(1)\right] \tag{46b}$$

Let $\mathcal{L}$ be set of indices for which $\hat{\alpha}_k = 1$. respectively. Then, for the policy $\hat{\bar{\alpha}}$, we have

$$\mathbb{E}\left[\sum_{k=0}^{M} R_k^{\mathrm{KL}}(\hat{B}_k, \hat{\alpha}_k, \hat{B}_{k+1})\right]$$
$$= -N\beta_C + \mathbb{E}\left[\sum_{k\in\mathcal{L}} \log \hat{B}_k(1)\right] + \mathbb{E}\left[\sum_{k\notin\mathcal{L}} \log \hat{B}_k(1)\right] \tag{47}$$

Since the belief vector only changes only when $\alpha_k = 1$, the sets $\{\tilde{B}_k\}_{k=0}^{N-1}$ and $\{\hat{B}_k\}_{k\in\mathcal{L}}$ are equal in expectation, so the first two terms in (46b) and (47) are equal. Thus, we only need to compare $\mathbb{E}\left[\sum_{k=N}^{M} \log \tilde{B}_k(1)\right]$ and $\mathbb{E}\left[\sum_{k\notin\mathcal{L}} \log \hat{B}_k(1)\right]$.

For $k < N$, by assumption we have $\tilde{B}_{k+1}(1) \geq \tilde{B}_k(1)$ in expectation, since $\tilde{\alpha}_k = 1$. In addition, for $k \geq N$ we have $\tilde{B}_k = \tilde{B}_N$, since $\tilde{\alpha}_k = 0$. Thus, we have that for all $k < N, k' \geq N$, $\tilde{B}_k \leq \tilde{B}_{K'} = \tilde{B}_N$ in expectation.

Since with policy $\hat{\bar{\alpha}}$, $\hat{\alpha}_k = 0$ for some $k < N$, it must be the case that $\hat{\alpha}_k = 1$ for some $k \geq N$. Let the largest K such that $\hat{\alpha}_k = 1$ be denoted K', and we have $K' \geq N$. Then, we have that $\hat{B}_{K'} = \tilde{B}_N$ in expectation. This implies that $\hat{B}_k \leq \tilde{B}_N$ in expectation for all $k < K'$, and in particular for all $k < K'$ such that $k \in \mathcal{L}$.

Therefore,

$$\mathbb{E}\left[\sum_{k\notin\mathcal{L}} \log \hat{B}_k(1)\right] \leq \mathbb{E}\left[\sum_{k=N}^{M} \log \tilde{B}_k(1)\right]. \tag{48}$$

■