Predictive Runtime Monitoring for Mobile Robots using Logic-Based Bayesian Intent Inference

Hansol Yoon and Sriram Sankaranarayanan .

***Abstract*— We propose a predictive runtime monitoring framework that forecasts the distribution of future positions of mobile robots in order to detect and avoid impending property violations such as collisions with obstacles or other agents. Our approach uses a restricted class of temporal logic formulas to represent the likely intentions of the agents along with a combination of temporal logic-based optimal cost path planning and Bayesian inference to compute the probability of these intents given the current trajectory of the robot. First, we construct a large but finite hypothesis space of possible intents represented as temporal logic formulas whose atomic propositions are derived from a detailed map of the robot's workspace. Next, our approach uses real-time observations of the robot's position to update a distribution over temporal logic formulae that represent its likely intent. This is performed by using a combination of optimal cost path planning and a Boltzmann noisy rationality model. In this manner, we construct a Bayesian approach to evaluating the posterior probability of various hypotheses given the observed states and actions of the robot. Finally, we predict the future position of the robot by drawing posterior predictive samples using a Monte-Carlo method. We evaluate our framework using two different trajectory datasets that contain multiple scenarios implementing various tasks. The results show that our method can predict future positions with precisely and efficiently, so that the computation time for generating a prediction is a tiny fraction of the overall time horizon.**

I. INTRODUCTION

Detecting and preventing imminent property violations is an important problem for the safe operation of autonomous robots in highly dynamic environments. Such violations include collisions between multiple robots, failure to respond to events or robots entering restricted areas. Detecting such violations at design time is often impractical: behaviors are dependent on possible environmental conditions. The space of possible behaviors is too large, or may not be completely known to the designers. Thus, runtime monitoring approaches have recently gained popularity. However, these approaches require a model of the robot's motion to predict its future position. Recent approaches have employed such models to detect the possible positions that a robot can reach in the near future using physics-based dynamic models and reachability analysis [1]–[4]. Similarly, a pattern-based approach predicts future positions based on historical data and predicts likely future positions [5] (Cf. [6] for a survey of trajectory prediction for dynamic agents).

However, forecasting future moves by extrapolating the past trajectories is often likely to fail unless we also have a specification of the task (or current subtask) that a robot is performing. In this paper, we term this as the robot's (current/short term) *intent*. This approach assumes that the robot has a high-level mission or intent. Furthermore, the robot is assumed to choose an "efficient" plan for implementing the mission. The efficient path can be either the shortest path or the *almost* shortest path. In many scenarios, this assumption is reasonable since operators want robots to implement more missions with limited resources. Therefore, if a robot does not choose an efficient strategy for completing a task, we can deduce that either the robot is not rational or that our current model of the robot's goals are incorrect [7]–[13].

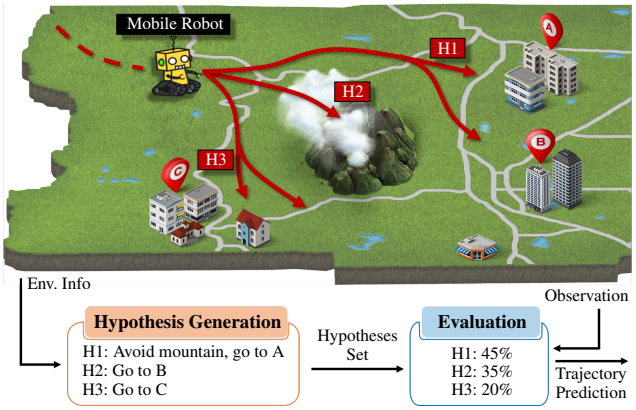

Fig. 1. The proposed Bayesian intent inference approach first generates a hypotheses set (H1-H3) and uses the robot's recent positions to update the probability of various hypothesized intents (H1-H3). Finally, we can predict likely future positions of the robot using the inferred distribution over possible intents.

In this paper, we use the robot's intent information for predictive runtime monitoring. Our assumption is that, if a robot has a high-level mission, we are able to (1) not only infer the mission by observing its behavior (2) but also use the information to predict future positions. Therefore, the ability to find the intent is key in our work.

To that end, we use temporal logic formulas to represent the intent. Temporal logics have been quite popular for specifying missions in a precise manner and generating efficient plans for carrying them out [14]–[20]. Temporal logics have been demonstrated as suitable reprentations of complex real-world missions such as surveillance and package delivery [18], [21]–[23]. We identify a subset of temporal logic formulas corresponding to the safety and guarantee formulas in the Manna-Pnueli hierarchy of temporal logic formulas [24]. Such formulas can be satisfied or violated by a finite prefix of an infinite sequence of actions and thus

Department of Computer Science, University of Colorado Boulder, USA, {firstname.lastname}@colorado.edu

quite suitable as representations of "near-term"/"immediate" intents suitable for finite time horizon predictions of the robot's position. The Bayesian intent inference framework then generates a finite set of possible intents using given patterns of temporal logic formulas and places a prior distribution on these formulas to represent the probability that a given formula represents the robot's intent. Next, we use a model of "noisy rationality" to provide a probability that a robot takes a given action in the workspace given its true intent. This model compares the cost of the action and the most efficient path from the resulting state to the overall goal of the intent against other possible actions. We use temporal logic planning techniques based on converting formulas to automata and solving shortest path problems to compute these costs.

Temporal logic specification inference from observation data have been studied widely in the recent past [25]–[28]. The main difference from our work is that they assume the entire trajectory is available at once, whereas we use the parts of the trajectory. Furthermore, our approach uses intents as a means to perform predictions of future positions.

We evaluate our framework on two datasets: a probabilistic roadmap simulation dataset, wherein we use the popular PRM planning technique to generate motion plans for some tasks while using our intent inference technique to predict the intents and future positions without knowledge of the overall mission plan. A second data set consists of trajectories of humans inside a room, called THÖR [29]: here we are provided noisy position measurements with unknown intents. Thus, both datasets include a moving agent implementing various subtasks on the way to a goal, which is unknown to our monitor. The results show that our method can predict future positions with high accuracy, and all computations can be implemented in real-time.

The contributions of this paper are as follows:

1) We introduce a Bayesian intent inference framework leveraging an intent information of a robot. The framework computes the probability distribution of all possible intents written in LTL.
2) Using the outputs of the framework, we can effectively carry out predictive monitoring that can be used in many robotic applications.
3) All computations can be implemented with sufficient efficiency to enable real-time monitoring.

To the best of our knowledge, this work is the first attempt to use a logic-based Bayesian intent inference for predictive monitoring.

II. Problem Formulation

Central to our framework is a "map" of the robot's workspace that is discretized into finitely many cells. Each cell is labeled with an atomic proposition that characterizes the attributes of the cell. We use the mathematical model of a *weighted finite transition system* to capture the map (or the workspace) of the robot.

Definition 1 (Weighted Finite Transition System): A weighted finite transition system $\mathscr{T}$ is a tuple (C, R, Π, L, ω) wherein C is a finite set of cells, $R \subseteq C \times C$ is the transition relation that represents all allowable moves from one cell to the next by the robot, Π is a set of boolean atomic propositions, $L : C \to 2^{\Pi}$ is a labeling function that associates each cell $c \in C$ with a set of atomic propositions $L(c)$, and $\omega : R \to \mathbb{R}_{\geq 0}$ maps each edge in R to a non-negative weight.

Therefore, the position of a robot at time t can be defined as a cell $\mathbf{x}_t \in C$. Atomic propositions label attributes/features such as *airport*, *fire*, *mountain*, and so on (see Fig. 1). A *path* in $\mathscr{T}$ is an infinite sequence of cells $p = c_0 c_1 c_2 \cdots$ such that $c_i \in C$ and $(c_i, c_{i+1}) \in R$ for each $i \in \mathbb{N}$.

a) Linear Temporal Logic: In this paper, we assume that a robot has a high-level mission to implement before going to a goal location. For example, "$\mathbf{H_1}$: Visit π_1, π_2, and π_3 in some order", or "$\mathbf{H_2}$: Visit π_3 while avoiding π_5". To formally express such requirements, we use linear temporal logic (LTL) whose grammar is defined as follows:

$$\varphi ::= true \,|\, false \,|\, \pi \in \Pi \,|\, \neg\varphi \,|\, \varphi \wedge \varphi \,|\, \bigcirc\varphi \,|\, \varphi \mathscr{U} \phi \,.$$

In addition, two temporal operators, *eventually* ($\Diamond\varphi$: $true \mathscr{U} \varphi$) and *globally* ($\Box\varphi$: $\neg\Diamond\neg\varphi$) can be derived. The formula $\Box\varphi$ is satisfied if φ holds for all time and $\Diamond\varphi$ is satisfied if eventually at some point in time φ is satisfied. We refer the reader to standard texts for a detailed description of temporal logic and its applications [30], [31]. Using LTL, we can express the mission $\mathbf{H_1}$: $\Diamond\pi_1 \wedge \Diamond\pi_2 \wedge \Diamond\pi_3$ and $\mathbf{H_2}$: $\Diamond\pi_3 \wedge \Box\neg\pi_5$. Using LTL is beneficial because it is capable of describing complex missions clearly although some fundamental properties like *safety* ($\Box\neg\varphi$) and *reachability* ($\Diamond\varphi$) are mostly used for robot missions in many scenarios, and because it enables us to use *temporal logic motion planning* [14]–[20].

b) Assumptions: We assume full knowledge of the transition system $\mathscr{T}$ is available at any time. Also, if the map is updated in the case of dynamic scenarios, the new information is assumed to be available immediately. On the other hand, the robot's mission is assumed to be unknown but expressible as a temporal logic formula involving atomic propositions in the map.

In this paper, we investigate two problems — intent inference and predictive monitoring. Fig. 2 shows how these problems relate to each other in our proposed framework.

Intent Inference: Given a transition system $\mathscr{T}$ and the recent history of robot cells at time t, $\mathbf{x}_t, \mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-h}$, we wish to infer a distribution of likely intents$\{(\varphi_1, p_1), \ldots, (\varphi_n, p_n)\}$, wherein φ_i is a temporal logic formula involving atomic propositions Π, and $p_i \geq 0$ is its associated probability with $\sum_{i=1}^{n} p_i = 1$.

Predictive Monitoring: Given a distribution over intents, we wish to compute a distribution of future positions $\mathbf{x}_{t+k}$ at time $t+k$. At time $t+1$, our approach receives new robot position $\mathbf{x}_{t+1}$, requiring updates to the intents, and the predicted future cell. This update needs to be computed in time that is much smaller than the overall sampling time.

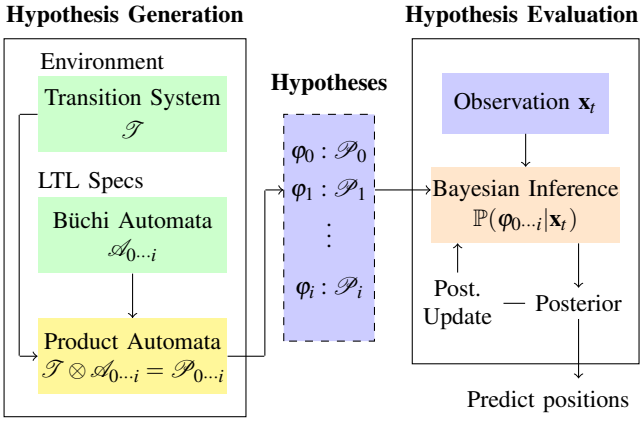

Fig. 2. Diagram of the Bayesian intent inference framework

III. Bayesian Intent Inference

We first introduce our Bayesian approach to solve the intent inference problem. The idea of our approach is to generate possible intents as our *hypotheses* and evaluate their probabilities using Bayesian inference (see Fig. 2).

A. Hypothesis Generation

Hypothesis generation is achieved using temporal logic *specification patterns* that have been explored in previous works (Cf. [16], [18]). Such patterns specify temporal logic formulae with "holes" that can be filled in with atomic propositions. Each such pattern defines a set of formulas obtained by substituting all possible atomic propositions of interest for each hole. To avoid potentially vacuous or inconsistent intents, we may further require that the same atomic proposition not be used in two distinct holes for a given template.

Example 1: We list some commonly encountered patterns of interest below. We substitute an atomic proposition in the place of a hole denoted by "$\boxed{?}$", ensuring that the same proposition does not appear in more than one hole.

- *Avoid Region*: $\Box\neg\boxed{?}$
- *Cover Region*: $\Diamond\boxed{?}$
- *First and Then Second Region*: $\Diamond\left(\boxed{?}\wedge\Diamond\left(\boxed{?}\right)\right)$
- *Reach While Avoid*: $\Diamond\boxed{?}\wedge\Box\neg\boxed{?}$

As a result, each pattern can be expanded out into a set of LTL formulae that represent possible intents of the agent.

B. Temporal Logic and Büchi Automata

We recall the standard connection between temporal logics and automata on infinite strings, specifically Büchi automata [32], [33]. Let φ be a temporal logic formula over atomic propositions in Π. Recall such a formula can be encoded as a nondeterministic Büchi automaton.

Definition 2 (Büchi Automaton): A Büchi automaton $\mathscr{A}$ is a tuple (Q,Π,E,q_0,F) wherein Q is a finite set of states; Π is a finite set of atomic propositions; $E\subseteq Q\times\Pi\times Q$ is a set of transitions, wherein each transition (q_i,π,q_j) indicates the transition from state q_i to q_j upon observing atomic proposition π; q_0 is an initial state and F is the set of accepting state.

Given an infinite sequence of atomic propositions $\pi_0,\pi_1,\pi_2,\ldots$, a run of the automaton is an infinite sequence of states $q_0,q_1,q_2,\ldots$, such that q_0 is the initial state and $(q_i,\pi_i,q_{i+1})\in E$ for all $i\geq 0$. Finally, a run is accepting iff it visits an accepting state $q\in F$ infinitely often. It is well-known that every LTL formula can be translated into a Büchi automaton [30], [31]. The problem of constructing a Büchi automaton from a LTL specification has been widely studied [34] with numerous tools such as SPOT [35].

a) Safety/Guarantee Formulas and Automata: In this paper, we focus on a very specific class of safety and guarantee formulas, originally introduced by Manna & Pnueli as part of a larger classification of all LTL formulas [24]. Briefly, safety formulas can be written using the $\Box$ operator with negations appearing only in front of atomic propositions, whereas guarantee formulas are written using the $\Diamond$ operator with negations appearing only in front of atomic propositions.

Example 2: Going back to the Example 1, we note that the "avoid regions" pattern is a safety formula, whereas the "cover regions" and "temporal sequencing" patterns are guarantee formulas. Note that the coverage with the safety pattern is the conjunction of a guarantee sub-formula (involving $\Diamond$) and a safety sub-formula (involving $\Box$).

Assumption: We will assume that any hypothesis being considered can be written as

$$\left(\bigwedge_{i=1}^{M}\Box\neg\pi_{s,i}\right)\wedge\left(\bigwedge_{j=1}^{N}\Diamond\pi_{g,j}\right), \tag{1}$$

wherein $A:\{\pi_{s,1},\ldots,\pi_{s,M}\}$ is disjoint from $B:\{\pi_{g,1},\ldots,\pi_{g,N}\}$, and $N>1$ (i.e, $B\neq\emptyset$). Such a formula represents the intent that the robot seeks to reach all regions labeled by atomic propositions in the set B, in some order, while avoiding all regions in A. More generally, however, our framework can accommodate the conjunction of safety formulas and guarantee formulas.

However, since our framework is *probabilistic* it associates a measure of belief/probability with each hypothesis. Also, since our framework is *dynamic*, these probabilities change over time. Thus, it is possible for our framework to implicitly infer a more complex high level objective that is not expressible in our restricted fragment of LTL. We will explore this aspect of our work further in the future.

We now consider a special type of Büchi automaton that we will call a *safety-guarantee* automaton.

Definition 3 (Safety-Guarantee Automaton): A Büchi automaton is said to be a safety-guarantee automaton if the set of states Q is partitioned into three mutually disjoint parts: $Q: Q_t\uplus F\uplus\{r\}$ wherein (a) the initial state $q_0\in Q_t\cup F$, (b) Q_t is a set of "transient" states such that no state in Q_t is accepting; (c) F is the set of accepting states, and (d) r is a special *reject state*. Furthermore, the outgoing edges from each state in F either take us to a state in F or to the reject state r. Finally, all outgoing edges from r are self-loops back to r. Fig. 3 illustrates safety-guarantee automata.

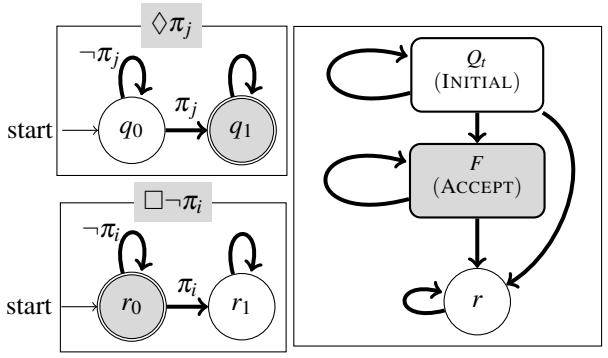

Fig. 3. **(left)** Büchi automata for $\Diamond\pi_j$ and $\Box\neg\pi_i$; and **(right)** Overall structure of a safety-guarantee automaton.

Lemma 1: A formula that satisfies the pattern in Eq. (1) is represented by a safety-guarantee Büchi automaton.

Proof: (Sketch) Note that such a formula is made up of a conjunction of $\Box(\neg\pi_j)$ and $\Diamond\pi_i$ subformulas whose automata are shown in Fig. 3 (left). The overall conjunction is represented by the product of these automata, wherein a product state is accepting iff each of the individual component states are accepting. The rest of the proof is completed by identifying the states in each partition to establish the overall safety-guarantee structure of the automaton. ■

b) Significance of Safety-Guarantee Structure: We will briefly explain why the overall structure of the automaton is important in our framework. Note that temporal logic formulas are quite powerful in expressing a variety of patterns that may include formulas such as $\Box\Diamond\pi$ which states that a cell satisfying the atomic proposition π must be reached infinitely often, or $\Diamond\Box\pi$ which states that the robot will eventually enter a region where π holds and stay in that region forever. Natually, it is *impossible* us to infer that such an intent holds or otherwise by observing any finite sequence of cells, no matter how long such a sequence may be. For instance, a robot intending to visit a region infinitely often may take a long time before its first visit to such a region since there are infinitely many steps ahead in the future. In this regard, the safety-guarantee structure allows the robot to signal its likely intent using a finite sequence: a robot intending to satisfy an intent can signal this in finitely many steps by reaching an accepting state in F. Likewise, a violation can also be seen in finitely many steps by reaching the reject state r.

c) Product Automaton: We define the Cartesian product between a weighted transition system $\mathscr{T}$ defining the workspace and a Büchi automaton $\mathscr{A}$.

Definition 4 (Product Transition System): The product automaton $\mathscr{T}\otimes\mathscr{A}$ is defined as the tuple: $(S,\delta,\hat{F},\hat{\omega})$:

1) S: $C\times Q$ is the Cartesian product of the set of cells in $\mathscr{T}$ and states in $\mathscr{A}$;
2) $\delta\subseteq S\times S$ is a transition relation s.t. $((c_i,q_i),(c_j,q_j))\in\delta$ iff $(c_i,c_j)\in R$ and $(q_i,\pi_k,q_j)\in E$ for some $\pi_k\in L(c_i)$;
3) $\hat{F}$: $C\times F$ is the set of accepting states, and
4) $\hat{\omega}((c_i,q_i),(c_j,q_j))$ is a weight function that is set to be equal to $\omega(c_i,c_j)$ if $((c_i,q_i),(c_j,q_j))\in\delta$

The product automaton models all the "joint" moves that can be made by a copy of the automaton $\mathscr{A}$ in conjunction with a transition system $\mathscr{T}$, wherein the atomic propositions labeling each cell in $\mathscr{T}$ governs the possible enabled edges in the automaton $\mathscr{A}$.

C. Cost of Formula Satisfaction

Let $\mathscr{T}$ be a weighted transition system describing the workspace of the robot and ψ be a formula that follows the pattern in Eq. (1), and described by a safety-guarantee automaton $\mathscr{A}_\psi$. For a given state $\mathbf{x}_t$ of $\mathscr{T}$, we define the cost of satisfaction: $\mathscr{C}(\mathbf{x}_t,\varphi)$ as the shortest path cost for a path in the transition system $\mathscr{T}$ whose atomic propositions satisfy the formula φ. Formally, we define (and compute) $\mathscr{C}(\mathbf{x}_t,\varphi)$ using the following steps:

1) Compute the product automaton $\mathscr{T}\otimes\mathscr{A}_\psi$.
2) Compute the shortest path cost from the product state $(\mathbf{x}_t,q_0)$ to the set of accepting states $\hat{F}$ in the product automaton, wherein the cost of a path is given by the some of edge weights along the path.

Note that the shortest cost path from a single product automaton state to a set of accepting states is defined as the minimum among all possible shortest path from the source to each element of the set. Since all edge weights are positive, we can calculate the cost from each cell $\mathbf{x}_t\in C$ to the set of accepting states in time using Dijkstra's algorithm (single destination shortest path). To handle a set of possible destination, we simply add a designated new destination node and connect all accepting states to it using a 0 cost edge. This calculation runs in time $O((|\delta|+|S|)\log(|S|))$ wherein $|S|=|C|\times|Q|$ is the number of states in the product automaton and $|\delta|=|R|\times|E|$ denotes the number of edges.

D. Bayesian Inference of Intent

Let $\mathscr{H}:\{\varphi_1,\ldots,\varphi_n\}$ be the set of hypothesized intents of the robot whose current cell is denoted by $\mathbf{x}_t\in C$. We will assume a prior probability distribution π over $\mathscr{H}$ wherein $\pi(\varphi_j)$ denotes the prior probability over hypothesis formula φ_j. Our initial prior starts out by assigning each hypothesis a uniform probability. The posterior from step $t-1$ forms the prior for step t with some modifications.

At each step, we obtain an updated robot position $\mathbf{x}_{t+1}\in C$ and use this fact to update the current distribution over $\mathscr{H}$. To do so, we require a model of robot decision making that determines the conditional probability $\mathbb{P}(\mathbf{x}_{t+1}\mid\mathbf{x}_t,\varphi)$: the probability given the intent φ and current cell $\mathbf{x}_t$, the robot moves to cell $\mathbf{x}_{t+1}$. We will make an assumption of *Boltzmann noisy rationality* [36].

a) Boltzmann Noisy Rationality Model: Let $\mathsf{next}(\mathbf{x}_t)$ denote all the neighboring cells to $\mathbf{x}_t$. We assume that for each cell $c\in\mathsf{next}(\mathbf{x}_t)$ the probability of moving to c is proportional an exponential of the sum of the cost of moving from $\mathbf{x}_t$ to c and the cost of achieving the goal from c.

$$\mathbb{P}(c|\mathbf{x}_t,\varphi_j)\ \propto\ \exp\left(-\beta(\omega(\mathbf{x}_t,c)+\mathscr{C}(c,\varphi_j))\right),$$

wherein β is a chosen positive number that represents the rationality. For $\beta=0$, the robot's choice is just a uniform

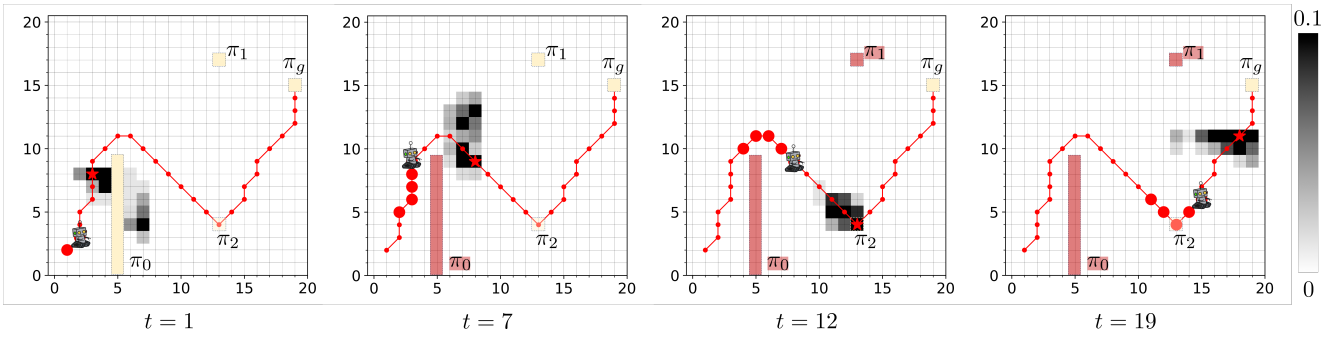

Fig. 4. Example scenario of predictive monitoring for the robot that has an underlying intent $\square\neg\pi_0 \wedge \Diamond\pi_2 \wedge \Diamond\pi_g$. The past five states (red circles) are used for Bayesian intent inference and our monitor computes a distribution of future states (gray). The future trajectory is shown using small dots and the ground truth of the future state is represented by red stars. As the robot navigates on the map, the monitor found π_0 and pi_1 are not a part of goals.

choice between all available moves regardless of the intent. However as $\beta \to \infty$, the agent simply chooses the optimal edge along the shortest cost path. With the appropriate normalizing constant, we note that

$$\mathbb{P}(\mathbf{x}_{t+1}|\mathbf{x}_t, \varphi_j) = \frac{\exp\left(-\beta\ \left(\omega(\mathbf{x}_t, \mathbf{x}_{t+1}) + \mathscr{C}(\mathbf{x}_{t+1}, \varphi_j)\right)\right)}{\sum_{c \in \mathsf{next}(\mathbf{x}_t)} \exp\left(-\beta\ \left(\omega(\mathbf{x}_t, c) + \mathscr{C}(c, \varphi_j)\right)\right)}. \tag{2}$$

Using Bayes rule, we can now compute the (unnormalized) posterior likelihood as follows:

$$\mathbb{P}(\varphi_j\ |\mathbf{x}_t, \mathbf{x}_{t+1}) \ \propto\ \mathbb{P}(\mathbf{x}_{t+1}\ |\ \mathbf{x}_t, \varphi_j)\ \times\ \pi(\varphi_j). \tag{3}$$

The posterior probability is calculated by normalizing this over all hypothesized intents $\varphi \in \mathscr{H}$. We will recursively update the prior at each step to yield the posterior at the next step. However, it is often useful to capture a change in the intent at each step by means of an "ε-transition":

$$\pi_{t+1}(\varphi_j) = (1-\varepsilon)\mathbb{P}(\varphi_j\ |\ \mathbf{x}_t, \mathbf{x}_{t+1}) + \varepsilon \frac{1}{|\mathscr{H}|}. \tag{4}$$

The so-called ε transition simply weights down the posterior by a factor $1-\varepsilon$ and adds a uniform probability distribution with a constant weight ε. This method allows us to quickly capture the new intent when an agent changes its intent during the operation. In our experiments, we fix $\varepsilon = 0.3$.

E. Posterior Predictive Distribution

Given the current posterior $\mathbb{P}(\varphi_j|\mathbf{x}_t, \mathbf{x}_{t+1})$ computed using Eq. (3), we wish to forecast the future position of the robot. We model the movement of the robot as a stochastic process:

1) Let initial position be $\mathbf{x}_{t+1}$ and the initial intent distribution be given by the distribution π_{t+1} from Eq. (4).
2) At time $t = t+k$, update the current intent distribution: $\pi_{t+k+1} = (1-\varepsilon)\pi_{t+k} + \varepsilon\ \mathsf{Uniform}(\mathscr{H})$, wherein $\mathsf{Uniform}(\mathscr{H})$ represents a uniform distribution over the elements of the finite set $\mathscr{H}$.
3) Sample an intent φ_j from π_{t+k+1}.
4) Sample $\mathbf{x}_{t+k+1}$ from the distribution $\mathbb{P}(\mathbf{x}_{t+k+1}|\mathbf{x}_{t+k}, \varphi_j)$ according to Eq. (2).

Using the procedure above, we obtain samples of potential future trajectories of the robot. We can use these trajectories to predict the possible future positions at a future time $t+T$. Notice, however, that our model implicitly assumes that the robot's intents change arbitrarily and are chosen afresh at each step according to the posterior distribution.

a) Example Scenario: Fig. 4 shows an example scenario of a robot in a workspace with four distinct regions labeled with atomic propositions π_0, π_1, π_2 and π_g as shown by the highlighted rectangles in the figure. The underlying (ground truth) intent is to avoid the region π_0, visit regions π_2 and π_g. The path taken by the robot is shown using the red circles whereas the predicted future distribution is shown using various shades of gray, the darker shade representing a higher probability. The ground truth future position is shown using the red star.

At time $t = 1$, having observed just two data points, the intent to avoid π_0 is guessed by our monitor. However, at time $t = 7$, the robot is seemingly unable to distinguish between the competing goals of reaching/avoiding π_1, π_g and π_2. However, at this time, the monitor predicts a right turn with a high probability though the robot's direction of travel would indicate that it continues moving in a straight line in the positive y direction.

At time $t = 12$, we see that the robot's direction of travel makes the intent to reach π_2 clear. Similarly, at time $t = 19$, we see that the goal of reaching π_1 or that of π_g are considered likely with π_g being seen as more likely to be the robot's intended target.

Thus, we see how the robot's future positions are predicted accurately by our monitor even though the set of possible intents are restricted to simple safety-guarantee formulas.

IV. Experimental Results

In this section, we evaluate the performance of our monitoring approach on two datasets – a synthetic data set generated by a Probabilistic Roadmap (PRM) motion planning algorithm that was used to plan paths satisfying randomly generated ground truth "intents", and the THÖR human trajectory dataset [29]. Using these, we will answer the following questions: (Q1) How does the prediction accuracy change as a prediction horizon increases? and (Q2) How does the computation time depend on the prediction time horizon?

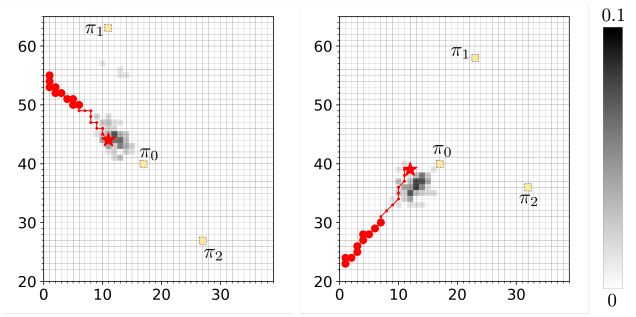

Fig. 5. Example of the evaluation setup showing goal regions labeled by atomic propositions and an underlying intent that seeks some subset of the regions/avoids the rest. The revealed trajectory is shown using red circles while the future trajectory is shown using dotted lines. Predicted distribution of future states is shown in various shades of gray with darker shades representing higher probabilities.

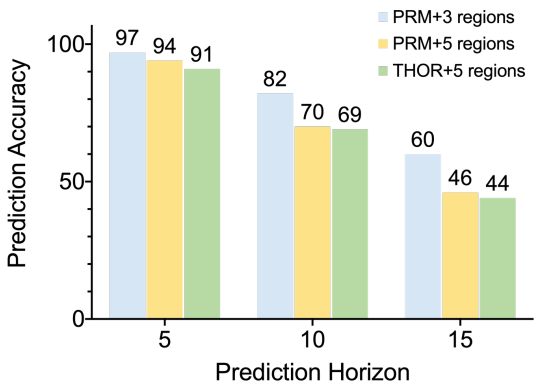

Fig. 6. Prediction accuracy tested on various settings.

TABLE I

COMPUTATION TIME FOR OUR APPROACH, AS RECORDED ON A MACBOOK PRO WITH 2.6 GHZ INTEL CORE I7 AND 16 GB RAM.

Map size	20×20	50×50	100×100
Product Automaton Construction (32 states in a Büchi automaton)	0.07	0.23	0.75
Bayesian Intent Inference with 32 hypotheses	0.16	0.56	2.13
300 Monte-Carlo Simulations (5 / 10 / 15 steps)	0.28 / 0.55 / 0.81		

Each evaluation is performed over a map with K distinguished regions marked by atomic propositions $\pi_1, \ldots, \pi_K$. The hypothesized intents consist of 2^K formulas, each of the form $\bigwedge_{j\in A} \Diamond \pi_j \ \wedge \ \bigwedge_{i\in B} \Box \neg \pi_i$ wherein $A \cap B = \emptyset$ and $A \cup B = \{1, \ldots, K\}$. Each evaluation consists of a path followed by the robot wherein the monitor predicts the probability distribution of the positions 5, 10 and 15 steps ahead. A prediction is deemed correct if the actual ground-truth position is predicted by our monitor as having a probability ≥ 0.01.

The trajectories are generated using two approaches:

PRM trajectories: We use a map with $N \times N$ grid cells and place K random regions on the map. Next, we mark a randomly chosen subset of these regions as obstacles and select the remaining regions as targets. We use the PRM motion planner off the shelf to generate a plan that may not necessarily be optimal, but is often close to being optimal. Our experiments vary $N \in \{20, 50, 100\}$ and $K \in \{3, 5\}$.

THÖR Human Motion Dataset: This publicly available dataset includes multiple human trajectory data recorded in a room of 8.4×18.8 meters [29]. The workspace consists of five goal locations around the room and one obstacle in the middle. Participants navigate between goals while avoiding the obstacle. To use this dataset for our monitor, we converted the workspace to a 50×50 grid map, and discretized human trajectory data. Among all trajectories, we selected 277 segments at random for our evaluation.

A. Evaluation of Accuracy and Computation Time

Fig. 6 shows that the prediction accuracy for the various datasets under varying prediction horizons. We note that the performance degrades as the prediction horizon grows, as expected. Nevertheless, our approach continues to provide useful information nearly 70% of the time on prediction horizons that are 10 steps ahead. Furthermore, this accuracy can be increased further by considering correlations of intents over time which is currently not performed in our approach. We also note that the accuracy degrades when more atomic propositions are considered and thus more hypothesized intents are available. Finally, we notice that the accuracy for the real-life human trajectory dataset is comparable with that of the synthesized PRM dataset. This partly validates the rationality hypothesis that underlies our work.

Next, we consider an evaluation of the computation time. The computational complexity of our approach dependes on the size of product automata and the number of hypotheses. Table I reports the computation time for constructing product automata, checking 32 intents, and Monte-Carlo simulations for various map sizes. We note that the computation times remain small even for a 100×100 grid and 32 intents each with 32 Büchi automaton states.

V. CONCLUSION

Thus, we have demonstrated a framework for inferring intents and predicting likely future positions of robots. Our framework can be extended in many ways including richer set of intents, alternative assumptions on how intents change over time, incorporating richer agent dynamics, maps with time-varying regions of interest, alternatives to the noisy rationality model considered, and finally, intents governing multiple agents. We propose to study these problems using the rich framework of logic, automata and games combined with fundamental insights from Bayesian inference and machine learning.

ACKNOWLEDGMENTS

This work was funded in part by the US National Science Foundation (NSF) under award numbers 1815983, 1836900 and the NSF/IUCRC Center for Unmanned Aerial Systems (C-UAS). The authors thank Prof. Morteza Lahijanian for helpful discussions.

REFERENCES

[1] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[2] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff, "Provably safe motion of mobile robots in human environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1351–1357.

[3] M. Koschi and M. Althoff, "Set-based prediction of traffic participants considering occlusions and traffic rules," *IEEE Transactions on Intelligent Vehicles*, 2020.

[4] Y. Chou, H. Yoon, and S. Sankaranarayanan, "Predictive runtime monitoring of vehicle models using bayesian estimation and reachability analysis," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[5] R. Peddi, C. D. Franco, S. Gao, and N. Bezzo, "A data-driven framework for proactive intention-aware motion planning of a robot in a human environment," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[6] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[7] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," in *Robotics: Science and Systems*, 2018.

[8] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance1," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020.

[9] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin, "A scalable framework for real-time multi-robot, multi-human collision avoidance," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 936–943.

[10] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5817–5823.

[11] B. I. Ahmad, J. K. Murphy, P. M. Langdon, and S. J. Godsill, "Bayesian intent prediction in object tracking using bridging distributions," *IEEE transactions on cybernetics*, vol. 48, no. 1, pp. 215–227, 2016.

[12] I. Hwang and C. E. Seah, "Intent-based probabilistic conflict detection for the next generation air transportation system," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2040–2059, 2008.

[13] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *Journal of guidance, control, and dynamics*, vol. 30, no. 2, pp. 370–382, 2007.

[14] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2689–2696.

[15] ——, "Motion planning with hybrid dynamics and temporal goals," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 1108–1115.

[16] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[17] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE transactions on robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[18] L. R. Humphrey, E. M. Wolff, and U. Topcu, "Formal specification and synthesis of mission plans for unmanned aerial vehicles," in *2014 AAAI Spring Symposium Series*, 2014.

[19] C. I. Vasile and C. Belta, "Reactive sampling-based temporal logic path planning," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4310–4315.

[20] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus, "Optimality and robustness in multi-robot path planning with temporal logic constraints," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 889–911, 2013.

[21] V. L. Somers and I. R. Manchester, "Priority maps for surveillance and intervention of wildfires and other spreading processes," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 739–745.

[22] G. Ferri, P. Stinco, G. De Magistris, A. Tesei, and K. D. LePage, "Cooperative autonomy and data fusion for underwater surveillance with networked auvs," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 871–877.

[23] S. Choudhury, K. Solovey, M. J. Kochenderfer, and M. Pavone, "Efficient large-scale multi-drone delivery using transit networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4543–4550.

[24] Z. Manna and A. Pnueli, "A hierarchy of temporal properties," in *Proc. Principles of Distributed Computing (PODC)*. ACM, 1990, p. 377–410.

[25] A. Shah, P. Kamath, J. A. Shah, and S. Li, "Bayesian inference of temporal task specifications from demonstrations," in *Advances in Neural Information Processing Systems*, 2018, pp. 3804–3813.

[26] J. Kim, C. Muise, A. Shah, S. Agarwal, and J. Shah, "Bayesian inference of linear temporal logic specifications for contrastive explanations." in *IJCAI*, 2019, pp. 5591–5598.

[27] M. Vazquez-Chanlatte and S. A. Seshia, "Maximum causal entropy specification inference from demonstrations," in *International Conference on Computer Aided Verification*. Springer, 2020, pp. 255–278.

[28] M. Vazquez-Chanlatte, S. Jha, A. Tiwari, M. K. Ho, and S. Seshia, "Learning task specifications from demonstrations," in *Advances in Neural Information Processing Systems*, 2018, pp. 5367–5377.

[29] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 676–682, 2020.

[30] Z. Manna and A. Pnueli, *Temporal Verification of Reactive Systems: Specification*, New York, 1992.

[31] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[32] P. Wolper, "Constructing automata from temporal logic formulas: a tutorial," in *Lectures on formal methods and performance analysis: first EEF/Euro summer school on trends in computer science*. Springer, 2002, pp. 261–277.

[33] W. Thomas, "Automata on infinite objects," in *Handbook of Theoretical Computer Science: Volume B: Formal Models and Semantics*, J. van Leeuwen, Ed. Amsterdam: Elsevier, 1990, pp. 133–191.

[34] P. Gastin and D. Oddoux, "Fast ltl to büchi automata translation," in *International Conference on Computer Aided Verification*. Springer, 2001, pp. 53–65.

[35] A. Duret-Lutz, A. Lewkowicz, A. Fauchille, T. Michaud, E. Renault, and L. Xu, "Spot 2.0 — a framework for LTL and ω-automata manipulation," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2016, pp. 122–129.

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