Stochastic Communication and Motion Planning via Learned Abstract World Representations

Mehdi Dadvar[1], Keyvan Majd [1], Georgios Fainekos[2], and Siddharth Srivastava[1]

***Abstract*— The increasing deployment of robots alongside humans necessitates sophisticated communication and motion planning to ensure safety and task achievability in social navigation scenarios. Existing methods often rely heavily on historical data and extensive expert hand-coding, which limits their scalability and generalizability. This paper introduces a novel framework that models social navigation as a Markov Decision Process (MDP), utilizing Conditional Abstraction Trees (CATs) to learn dynamic abstract world representations and policies that focus on critical aspects of interaction. In the offline phase, the framework operates within a simulator, while in the online phase, it deploys the learned representations and policies in real-world scenarios for ongoing refinement and adaptation. Integral to our approach is a Dynamic Bayesian Network (DBN) based human sensor and belief model that accounts for humans' imperfect perception to enhance the prediction of human motion. We evaluated our method through extensive simulations and user studies involving physical experiments, demonstrating its effectiveness in managing critical interactions and ensuring safety and task completion across various scenarios.**

I. Introduction

The increasing integration of robots into human environments has highlighted the critical need for explicit communication [1] to ensure safe and efficient robot performance, especially in close proximity to humans [2], [3], [4]. Previous work [5] has laid the groundwork for joint communication and motion planning, but two significant issues remained unresolved. First, the planning process must account for the highly stochastic nature of environments where human presence introduces uncertainty, and robot actions themselves are inherently stochastic. Second, the inclusion of robot communication actions and human physical and mental states significantly expands the planning space, making it computationally intractable. This paper aims to develop a planning scheme that not only jointly optimizes the robot's communication and navigational actions in stochastic settings but also automatically learns an abstract world representation, enhancing its ability to handle stochasticity and improve scalability.

Robot motion planning has been extensively studied in the robotics field [6], [7], yet the introduction of human presence into the environment significantly jeopardizes the safety and task achievability of the interaction [8], [5]. Predicting human motion in relation to robot actions is essential but presents a complex and critical challenge. Some studies have developed human prediction models leveraging historical data [9], [10], planning methods [11], [12], or the geometrical aspects of natural human motion [13], [14] to forecast human movement. However, these approaches often overlook the influence of robot actions and communications on human behavior, as well as the imperfect nature of human observation. This oversight severely compromises the ability of planning frameworks to effectively predict human motion, leading to undesirable situations such as deadlocks or unsafe encounters.

While employing a human prediction model is essential, it significantly expands the planning state space by incorporating the human's physical and mental states, which can render the planning process intractable or computationally inefficient. This complexity highlights the need for abstract world representations in robot planning [15]. However, conventional abstraction methods that create uniform or static representations are inadequate for dynamic environments [16]. Ideally, an abstraction should dynamically provide higher precision around salient parts of the state space, such as when the robot is in close proximity to a human or engaged in a critical subtask, and remain coarse in less critical areas. Our previous work has introduced conditional abstraction trees to address this issue [17], starting with a coarse abstraction and refining it as the agent interacts with the world. Yet, the high levels of stochasticity and the dynamic nature of human-robot interactions can still overwhelm this abstract refinement process, leading to overly detailed representations that defeat the purpose of abstraction. This identifies another significant and critical gap in the literature for developing a sophisticated, automatically constructed abstract world representation that effectively captures the dynamic nuances of human-robot interaction despite the highly stochastic environment.

In this work, we model social navigation as a Markov Decision Process (MDP) with a dual-phase approach: offline and online. During the offline phase, we focus on learning an abstract world representation alongside an abstract policy for the robot's communication and motion, employing Conditional Abstraction Trees (CATs). This phase necessitates a robust modeling of human behavior, for which we introduce a Dynamic Bayesian Network (DBN) based sensor model and belief model of humans. This model account for the stochastic behavior of humans, including their imperfect and noisy observations and varying levels of attentiveness during interactions, thereby enhancing the accuracy of human behavior predictions within the simulator.

In the online phase, we deploy the learned policy and

*This work was not supported by any organization

[1]School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, USA mdadvar@asu.edu

[2]Toyota Motor North America, Research and Development, Ann Arbor, MI, USA

abstract world representation in a real-world scenario. This deployment allows the planning framework to continue updating the policy and refining the world representation dynamically. To optimize the learning of the abstract world representation and policy, we introduce the concept of local and global CATs. Locally, we learn a CAT focusing on immediate information relevant to the robot, such as potential collisions with humans and other safety aspects of the interaction. Globally, we learn a CAT that addresses broader interaction dynamic like the robot's goals and task accomplishments. These CATs are designed to interact dynamically throughout both phases, enabling an effective capture of the nuances of human-robot interaction. User studies and subsequent results confirm the effectiveness of our method, demonstrating significant improvements in the safety and efficiency of human-robot interactions.

II. Related Work

The development of human-robot communication in social navigation involves varying strategies from implicit to explicit communication, each with distinct challenges. Although various methods have been studied for robots to implicitly communicate and influence human behavior [18], environmental constraints can sometimes limit the robot's physical actions, making them inadequate for implicit communication. This underscores the necessity for integrating explicit communication alongside implicit methods. To address this, a method presented in [19] employs large language models (LLMs) to create adaptable robot motion based on social contexts. This approach faces limitations in generalizability and depends heavily on precise hand-coded prompts, struggling with the translation of numerical world orientations into text.

Another group of methods employs more sophisticated formulations for communication planning via MDP and Partially Observable Markov Decision Process (POMDP) frameworks. For instance, the method presented in [20] uses inverse reinforcement learning to develop a human model, which it then leverages to optimize robot controls alongside communication actions. However, this approach overlooks the imperfect perception between humans and robots, assuming flawless reception of communications and unrealistically granting humans access to the robot's future trajectories. Similarly, the method proposed in [21] utilizes an Agent Markov Model within a framework modeled as an POMDP. While this method accounts for the robot's imperfect perception, it does not adequately address humans' imperfect observations. Furthermore, like the approach in [20], it relies on historical data and extensive domain expert involvement to account for the effects of robot communication on learned human models, making these methods less scalable and inapplicable in scenarios where such data is unavailable. In contrast, the method presented in [22] introduces a framework that relies less on historical data, using common probabilistic robotics representations to dictate communication content and timing. However, this approach does not jointly optimize the robot's navigational and communication actions and primarily focuses on information acquisition rather than influencing human behavior for safety. It accounts for the robot's imperfect observations but neglects the human's, a critical oversight in ensuring safety in social navigation.

Our work integrates both implicit and explicit communication with joint motion planning, while explicitly accounting for the noisy and imperfect observations of human behavior. Although our approach models social navigation as an MDP, it does not depend on the availability of historical interaction data. Instead, it utilizes interactions with a simulator during the offline phase. Furthermore, our method requires minimal to no expert input as the DBN parameters of human sensor model are the only inputs needed and can be autonomously learned. Moreover, our approach learns the robot's policy and an abstract world representation on-the-fly, thereby enhancing scalability and generalizability. While previous efforts have explored abstraction in robot planning, none have dynamically constructed conditional abstractions in real-time as our method does. To the best of our knowledge, this work is the first to propose a versatile robot planning framework using automatically learned abstraction that effectively addresses the nuances, dynamics, and stochasticity of human-robot interactions through the innovative use of local and global CATs.

III. Background

A. Markov Decision Processes

MDPs are defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces. The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ specifies the probabilities of transitioning between states, while the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ assigns a reward to an action taken in a specific state. γ is the discount factor that values future rewards. A policy π, which maps states to actions, aims to maximize the expected cumulative reward [23], [24]. Given the potential complexity of the state space, state abstraction reduces the state space dimensionality, mapping $\mathcal{S}$ to a reduced abstract state space $\bar{\mathcal{S}}$ through an abstraction function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ [25], [26]. As the result of of the state abstraction, the abstract MDP is defined as $\bar{M} = \langle \bar{\mathcal{S}}, \mathcal{A}, \bar{\mathcal{T}}, \bar{\mathcal{R}}, \gamma \rangle$ derived via ϕ. The abstract transition function $\bar{\mathcal{T}}$ and reward function $\bar{\mathcal{R}}$ aggregate transitions and rewards from the concrete states, weighted by a function $w(s)$, where $\sum_{s \in \phi^{-1}(\bar{s})} w(s) = 1$ and $w(s) \in [0, 1]$:

$$\bar{\mathcal{R}}(\bar{s}, a) = \sum_{s \in \phi^{-1}(\bar{s})} w(s) \mathcal{R}(s, a), \tag{1}$$

$$\bar{\mathcal{T}}(\bar{s}, a, \bar{s}') = \sum_{s \in \phi^{-1}(\bar{s})} \sum_{s' \in \phi^{-1}(\bar{s}')} w(s) \mathcal{T}(s, a, s'). \tag{2}$$

B. Conditional Abstraction Trees

In many real-world domains, such as social navigation, states are naturally expressed in terms of values of different variables. One method to construct abstraction is by partitioning the range of each state variable. Unlike trivial state abstraction, conditional abstraction considers the abstraction of one state variable contingent on the values of other state

variables [17]. As a result, the abstractions become dynamic, adjusting as the agent's state changes. A CAT structures these conditional relations hierarchically. At its root, the CAT begins with a coarse abstraction that encompasses a complete range for each state variable. This initial coarse abstraction is subsequently refined at the deeper levels of the tree, where the leaf nodes represent the abstract states. Given a CAT, the abstraction function $\phi(s) : \mathcal{S} \rightarrow \bar{\mathcal{S}}$, as detailed in Section III-A, is a level-order tree search starting from the root. Below, we provide the formal definitions of CATs, the refinement function, and the search process.

Generally, a concrete state $s \in \mathcal{S}$ can be defined as a set of n state variables such that $\mathcal{V} = \{v_i | i = 1, \ldots, n\}$. So an abstraction node in a CAT is defined as $\Theta = \{\theta_i | i = 1, \ldots, n\}$, where θ_i is the the partitioned range for state variable v_i. Each abstraction node can be refined w.r.t. a state variable v_i through a function $\delta(\Theta, i, f)$, which divides the range of the partition θ_i into f equal sub-ranges.

Definition 1: Let $\Theta = \langle \theta_1, \ldots, \theta_n \rangle$ be an abstract state for a domain with variables $v_1, \ldots, v_n$. We define the f-split refinement of Θ w.r.t. variable i as $\delta(\Theta, i, f) = \{\Theta^1, \ldots, \Theta^f\}$, where $\theta_i = [l, h]$ is partitioned into f new boundaries at least $\|\theta\|/f$ values apart.

Next, we must define the relational properties between two abstraction nodes, Θ_a and Θ_b, to establish whether one abstraction is a refinement of the other.

Definition 2: Given a set of abstractions Ψ, for $\Theta_a, \Theta_b \in \Psi$, Θ_b is considered a refinement of Θ_a, denoted $\Theta_b \triangleright \Theta_a$, if for every $i \in [1, n]$, $\theta_i^b \subseteq \theta_i^a$ and Θ_b is a direct refinement of Θ_a, denoted $\Theta_b \trianglerighteq \Theta_a$, if $\exists\, i\ (\theta_i^b \subset \theta_i^a)$ and for all $k \neq i$, $\theta_k^b = \theta_k^a$.

We can now formally establish a CAT as a tree structure, denoted by ξ, that constructs and maintains a hierarchy of conditional partitions represented by Θ.

Definition 3: A conditional abstraction tree (CAT) is defined as $\xi = \{N, E\}$, where N is the set of nodes and E is the set of edges. Each node in N corresponds to an abstraction Θ, s.t. $N = \{\Theta_m | m \in [1, n_\xi]\}$, where n_ξ is the cardinality of CAT and the root node of the tree is the initial abstraction Θ_{init}. Every parent Θ_p and child Θ_c nodes in ξ are connected via an edge e_p^c s.t. $e_p^c \implies \Theta_c \trianglerighteq \Theta_p$. $L_\xi = \{\Theta_m | (\forall k \in [1, n_\xi])(\Theta_k \not\trianglerighteq \Theta_m)\}$ is defined as the set of leaf nodes representing the set of abstract states.

IV. Problem Formulation

We formulate the social navigation problem as an MDP at the concrete level. This formulation captures the dynamics and interactions between human and robot agents within a shared environment. Here are the key components of our MDP:

- **State Space** $\mathcal{S}$: The state space is divided into two subsets: $\mathcal{S}_H$ for the human state variables and $\mathcal{S}_R$ for the robot state variables. Thus, the complete state space is expressed as $\mathcal{S} = \mathcal{S}_H \times \mathcal{S}_R$, encompassing all variables that define the positions, orientations, and other relevant statuses of both humans and the robot.

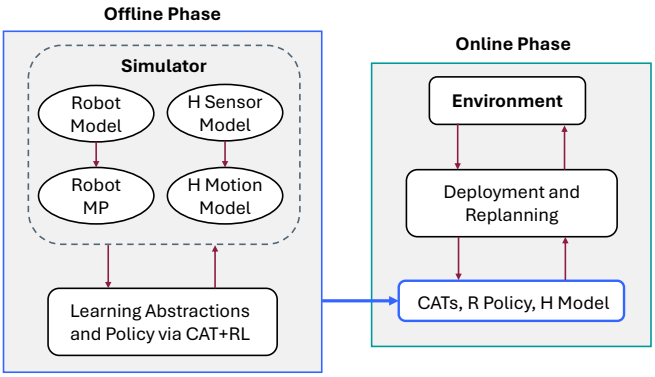

Fig. 1. Overview of the proposed approach: The **offline phase** involves interaction with a simulator featuring a robot model, motion planner, and a DBN-based human sensor model for learning abstractions and policy. The **online phase** depicts the deployment of learned CATs and policy in real-world settings, allowing continual adaptation based on new interactions.

- **Action Space** $\mathcal{A}$: The action space $\mathcal{A}$ encompasses the robot's actions, which are unified as $\mathcal{A} = \mathcal{A}_\pi \cup \mathcal{A}_c$. Here, $\mathcal{A}_\pi$ denotes navigational actions, and $\mathcal{A}_c$ represents communication actions.
- **Transition Model**: The overall state transition probabilities are given by the product of human-related transitions $\mathcal{T}_H(s'_H \mid s, a)$ and robot-related transitions $\mathcal{T}_R(s'_R \mid s, a)$, represented as $\mathcal{T}(s' \mid s, a) = \mathcal{T}_H(s'_H \mid s, a) \cdot \mathcal{T}_R(s'_R \mid s, a)$.
- **Reward Function** The reward function, $R(s, a)$, assigned to the robot at each timestep, penalizes unnecessary robot movements and close proximity to humans to ensure safety, while encouraging task accomplishment for both the human and the robot.
- **Discount Factor**: The discount factor, denoted by $\gamma \in [0, 1]$ quantifies the preference for immediate rewards over future rewards.

The solution to this MDP is defined by the robot policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$, dictating the robot's action. The policy π integrates both navigational actions $\mathcal{A}_\pi$ and communication actions $\mathcal{A}_c$. $\mathcal{A}_\pi$ in this context refers to high-level navigational actions, such as transitioning from one abstract region to another. To execute these high-level actions, a robot motion planner is employed, which is integrated into the overall robot policy. This integration ensures the downward refinability of the robot's high-level navigational ractions, while optimizing the robot's policy π.

V. Our Approach

A. Overview

The goal of this research is to develop a robust method that enables the learning of local and global conditional abstractions of world representation, as well as an abstract policy for robot navigational actions and communications in stochastic social navigation contexts. As Fig. 1 illustrates the overall procedure of our proposed approach, our approach is structured into two distinct phases: offline and online.

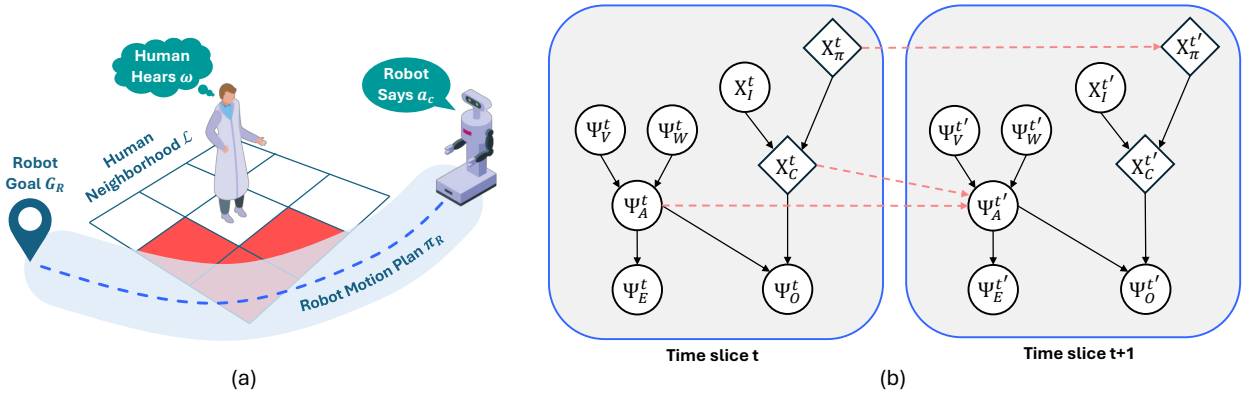

Fig. 2. Probabilistic Human Sensor Model and Belief Inference. (a) An example illustrating the interaction between a human and a robot, where the effects of the robot's actions and communications are probabilistically modeled over the human's local neighborhood to infer the human's belief about the robot's future location. (b) The Dynamic Bayesian Network (DBN) designed as the human sensor model. In this DBN, circles represent random variables, while diamonds represent decision nodes.

During the offline phase, the robot engages with a simulator to learn the necessary abstractions and abstract policy. This simulation integrates various components, including a robot model with a motion planner and a human sensor and movement model. A key development in this phase is the development of a DBN based human sensor and belief model. This model probabilistically estimates human beliefs regarding the robot's future locations, which facilitates more accurate predictions of human movements.

In the online phase, the learned local and global CATs and the robot's abstract policy are deployed to manage interactions with the physical world. This phase allows the robot to continue updating its policy and abstractions based on new real-world interactions, ensuring continual improvement and adaptation. Additionally, the human model remains accessible during the online phase, providing opportunities to update the DBN parameters of the human model based on real-world samples.

B. Human Sensor Model

To model the stochastic behavior of a human in a social navigation scenario, it is essential to account for how the human observes the robot's actions and communication. This observation is further influenced by the human's attention toward the situation. By incorporating such a model, we can probabilistically infer the human's belief about the robot's future location. As depicted in Fig. 2 (a), we consider a discrete local neighborhood for the human. Let $\mathcal{L}_H$ be the set of these discretized zones $\{l_1, \ldots, l_\ell\}$. The goal is to probabilistically infer the human's belief for each region of their local neighborhood, given the robot's physical actions and communication signals.

We employ a DBN architecture to probabilistically model human observations, denoted as Ψ_O. Fig. 2(b) illustrates the influence diagram of this model, capturing the dependencies and interactions between the key variables. According to the diagram, human observation Ψ_O is influenced by two main factors: a) the robot's communication signals χ_C, which is a decision node influenced by the robot's motion plan node χ_π and the intersection of its future path with the human's local neighborhood χ_I; and b) human attention Ψ_A, which is affected by the human's view of the scenario Ψ_V, cognitive workload Ψ_W, and the robot's communication from the previous time slice.

The probability distribution over Ψ_A^{t+1} can be computed as $P(\Psi_A^{t+1} \mid \chi_C^t = x_C, \Psi_V^{t+1} = \psi_v, \Psi_W^{t+1} = \psi_W)$, using the samples x_C, ψ_v, and ψ_W. The human's view Ψ_V can be sampled based on the geometric configuration of the human, the robot, and environmental factors such as obstacles. The human's cognitive workload ψ_W can be sampled based on visual evidence, for example, if the human is engaged in a specific task such as looking at their phone. If samples of Ψ_V and ψ_W are unavailable, other indicators such as active eye contact Ψ_E between the human and the robot can serve as evidence of human attention Ψ_A. With a sample of Ψ_E, the probability distribution over Ψ_A can be inferred using logical filtering.

Given the samples over χ_C^{t+1} and Ψ_A^{t+1}, we can infer the probability distribution over Ψ_O^{t+1} as $P(\Psi_O^{t+1} \mid \chi_C^{t+1} = x_C, \Psi_A^{t+1} = \psi_A)$. The conditional probability distributions specified in the DBN reflect a design choice that can be adapted based on the specifics of the scenario. It can be manually defined based on domain knowledge or learned from data through interaction history between the robot and humans.

C. Human Belief

Once we could infer the probability distribution over human's observation Ψ_O, we can now infer the humans belief over robot's future location. In the example of Fig. 2 (a), the robot intends to go to its goal at G_R by following motion plan π_R. It's communication signal a_C, is perceived by human as ω which we modeled it through the inference tacks over Ψ_O. Now, we want leverage the estimate human observation to infer human's belief over robot's future location. To do so, we sample χ_C^{t+1}, biased to the estimated probability distribution over Ψ_O^{t+1}, denoted as $\breve{x}_C^{t+1}$. Then, we use the belief update formula below to infer the probability

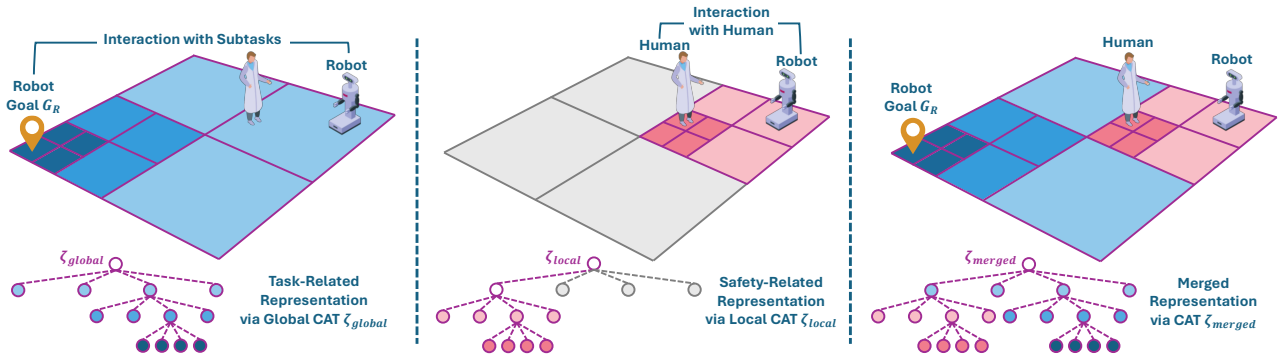

Fig. 3. This figure illustrates the interaction between local and global CATs. On the **left**, a global CAT is learned with respect to the TD-error dispersion caused by task-related rewards. In the **middle**, a local CAT is learned based on safety-related rewards. On the **right**, the conceptual merging of detailed information from the local CAT into the global CAT is shown, constructing a unified abstract world representation.

distribution over χ_I^{t+1}:

$$B(\chi_I^{t+1}) = \alpha P(\chi_C^{t+1} \mid \chi_I^{t+1}) \sum_{\chi_I^t} P(\chi_I^{t+1} \mid \chi_I^t) B(\chi_I^t). \quad (3)$$

Here, $B(\chi_I^{t+1})$ represents the updated belief for χ_I^{t+1}, $P(\chi_C^{t+1} \mid \chi_I^{t+1})$ is the likelihood of observing χ_C^{t+1} given χ_I^{t+1}, $P(\chi_I^{t+1} \mid \chi_I^t)$ denotes the transition model, $B(\chi_I^t)$ is the belief at the previous time step, and α is the normalization factor. This belief update inference is repeated independently for each region in the human's local neighborhood $\mathcal{L}_H$, ultimately yielding a probability distribution for every region, indicating whether or not the human believes the robot's future path will intersect with that region.

D. Human Movement Model

In a cooperative and rational social navigation scenario, a human is likely to avoid regions they believe the robot will occupy. Thus, we can represent the belief probability distribution for each region of the human's local neighborhood, $\mathcal{L}_H$, as an impassable region from the human's perspective. By sampling these impassable regions from the belief distribution $B(\chi_I^{t+1})$ for each region $l_i \in \mathcal{L}_H$, we can achieve a more accurate world representation from the human's viewpoint. Consequently, even with a deterministic underlying human prediction model, this updated world representation with impassable regions allows us to: 1) generate a probabilistic prediction of human movement due to the probabilistic modeling of human beliefs and impassable regions, 2) probabilistically model the effect of the robot's actions and communication on human behavior, and 3) remain independent of a specific human movement model, as a variety of movement models can be employed within the proposed framework.

The process of modeling the imperfect observation Ψ_O of the human over robot communication, deriving the human's belief $B(\chi_I^{t+1})$ about the robot's future location, incorporating the constraints of probabilistic impassable regions in the human's assumed world representation, and employing an underlying human prediction model, together explain how we derive the human transition model $\mathcal{T}_H(s'_H \mid s, a)$.

E. Local and Global CATs

Conditional abstractions construct a world representation based on samples gathered from the agent's interaction with the environment. Learning a CAT follows a repetitive three-phase process: 1) the **learning phase**, where a coarse abstraction is initialized, and an abstract policy is learned; 2) the **evaluation phase**, where the current abstraction is assessed by interacting with the environment using the fixed abstract policy; and 3) the **refinement phase**, where abstract states that exhibit instability during the evaluation phase are refined. It has been shown that the dispersion of TD-errors for an abstract state is an effective criterion for identifying unstable abstract states [17]. Dispersion can arise from broader interaction dynamics, such as the robot's goals and task accomplishments, or from immediate information relevant to the robot, like potential collisions with humans and other safety aspects of the interaction. Relying on both types of dispersion to create an abstract world representation via a single CAT can result in overly detailed abstractions. To address this, we introduce the concept of global and local CATs, which are simultaneously learned as the agent interacts with the environment.

The first step to achieving a separable dispersion of TD errors is to factorize the reward function at the concrete level such that $R(s,a) = R_{\text{task}}(s,a) + R_{\text{side}}(s,a)$. During the evaluation process, the agent may encounter the same abstract state $\bar{s}$ multiple times, resulting in a set of TD errors logged for each component of the reward function. The observed dispersion of TD errors under R_{task} is defined as $\Gamma_{task} = \{d_m^{task} \mid m \in [1, n_{\text{visited}}]\}$, where n_{visited} is the number of visited abstract states during the abstraction evaluation phase and d_m^{tasks} denotes the set of logged task-related TD errors for each visited abstract state. Similarly, TD errors under R_{side} is defined as $\Gamma_{side} = \{d_m^{side} \mid m \in [1, n_{\text{visited}}]\}$, where d_m^{tasks} denotes the set of logged safety-related TD errors for each visited abstract state. Let $\texttt{UnstableStates}(\Gamma)$ denote a function that identifies the set of unstable states in the form of node Θ in a CAT, based on Γ_{task} and Γ_{side}. For each visited abstract state in Γ_{task} and Γ_{side}, $\texttt{UnstableStates}$ calculates the maximum

Algorithm 1: Offline Phase of Learning

1: initialize ζ_{local} and ζ_{global}
2: **for** $episode = 1, n_{epi}$ **do**
3: $s \leftarrow$ `reset()`
4: **for** *steps* in *episode* **do**
5: $\zeta_{merged} \leftarrow \texttt{Merge}(s, \zeta_{local}, \zeta_{global})$
6: $\bar{s} \leftarrow \texttt{FindAbstract}(\zeta_{merged}, \Theta_{init}, s)$
7: $a_c, a_\pi \leftarrow$ Choose navigation and communication actions $\bar{\pi}(\bar{s})$
8: Translate a_c w.r.t. landmarks and transmit
9: $s', \bar{r}, done \leftarrow \texttt{ExecuteNavigation}(a_\pi)$
10: $\bar{s}' \leftarrow \texttt{FindAbstract}(\xi, \Theta_{init}, s')$
11: $\bar{\pi} \leftarrow \texttt{TrainPolicy}^{\bar{\pi}}(\bar{s}, \bar{s}', a, \bar{r})$
12: $s, \bar{s} \leftarrow s', \bar{s}'$
13: **if** $\bar{M}$ needs refinement **then**
14: $\Gamma_{task}, \Gamma_{side} \leftarrow \texttt{Evaluate}(M, \xi, \bar{\pi}, n_{eval})$
15: refine ζ_{local} and ζ_{global} based on Γ_{task} and Γ_{side}
16: **return** $\zeta_{global}, \zeta_{local}, \bar{\pi}$

normalized standard deviation (SD) of TD errors across all actions and returns the states whose normalized SD exceeds a given threshold μ_{SD}. During the refinement phase, unstable states identified in Γ_{task} are refined in the global CAT ζ_{global}, while unstable states identified in Γ_{side} are refined in the local CAT ζ_{local}. This refinement is performed using the refinement function δ, as defined in Sec. III-B, for each unstable abstract state in both ζ_{local} and ζ_{global}.

Although we have explained how both local and global CATs can be learned during the learning, evaluation, and refinement phases, we still require a mechanism to construct a unified CAT, referred to as the merged CAT ζ_{merged}. This merging is state-dependent, meaning that additional refinements from the local CAT ζ_{local} are only added to the global CAT ζ_{global} when the human and the robot are in close proximity, as shown in Fig. 3. In Fig. 3 (left), when R and H are in close proximity, a more refined local representation from the local CAT ζ_{local}, as shown in Fig. 3 (middle), is required to ensure that the robot's high-level actions are more precise around the human. The merge function is denoted as $\texttt{merge} : \mathcal{Z} \times \mathcal{Z} \times \mathcal{S} \rightarrow \mathcal{Z}$, where $\mathcal{Z}$ is the CAT space containing all possible conditional abstractions. To implement this function, we first need a criterion for defining close proximity. A suitable approach is to use the human's local neighborhood as the definition of close proximity. Thus, when R is within H's local neighborhood, the agents are considered to be in close proximity. Given a state s, if R and H are in close proximity, then we have two abstractions for the concrete state s under ζ_{local} and ζ_{global}, denoted as θ_{local} and θ_{global}, respectively. If θ_{local} is finer than θ_{global}, then, for each state variable, θ_{global} is refined using the refinement function δ until it reaches the same granularity as θ_{local}. The sub-tree added to θ_{global} as a result of these refinements is considered temporary and will be undone once the state s changes.

F. Overall Procedure

Algorithm 1 explains the overall procedure of our proposed method. This phase begins by initializing the local and global CATs, ζ_{local} and ζ_{global}, respectively (line 1). In each step of the episode, the local and global CATs are merged based on the current state s to form ζ_{merged} (line 5). The abstract state $\bar{s}$ is then determined using the `FindAbstract` function (line 6). The algorithm selects navigation and communication actions, a_π and a_c, from the abstract policy $\bar{\pi}$ (line 7), and the communication action is translated relative to the environment before being transmitted. The navigation action is executed via the underlying robot motion planner, resulting in the next state s', a reward $\bar{r}$, and a termination condition (line 9). The abstract policy is then updated using the `TrainPolicy` function (line 11). If model refinement is necessary (line 13), task- and safety-related dispersions (Γ_{task} and Γ_{side}) are evaluated, and the local and global CATs are refined accordingly (lines 14). We set the algorithm to check the recent success rate of the robot every n_{check} episodes where the refinement condition evaluates to true if the success rate is below some threshold t_{succ}. The algorithm completes by returning the refined abstractions $\zeta_{global}, \zeta_{global}$, and the learned abstract policy $\bar{\pi}$.

REFERENCES

[1] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023.

[2] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation.," in *Robotics: science and systems*, 2012.

[3] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *2013 IEEE international conference on robotics and automation*, pp. 2153–2160, IEEE, 2013.

[4] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, IEEE, 2017.

[5] M. Dadvar, K. Majd, E. Oikonomou, G. Fainekos, and S. Srivastava, "Joint communication and motion planning for cobots," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 4771–4777, IEEE, 2022.

[6] J.-C. Latombe, *Robot motion planning*, vol. 124. Springer Science & Business Media, 2012.

[7] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science robotics*, vol. 8, no. 84, p. eadf7843, 2023.

[8] K. Majd, S. Yaghoubi, T. Yamaguchi, B. Hoxha, D. Prokhorov, and G. Fainekos, "Safe navigation in human occupied environments using sampling and control barrier functions," *arXiv preprint arXiv:2105.01204*, 2021.

[9] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 336–345, 2017.

[10] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1349–1358, 2019.

[11] S. Huang, X. Li, Z. Zhang, Z. He, F. Wu, W. Liu, J. Tang, and Y. Zhuang, "Deep learning driven visual path prediction from a single image," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5892–5904, 2016.

[12] M. Shen, G. Habibi, and J. P. How, "Transferable pedestrian motion prediction models at intersections," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4547–4553, IEEE, 2018.
[13] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th international conference on computer vision*, pp. 261–268, IEEE, 2009.
[14] J. v. d. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.
[15] N. Shah, "Autonomously learning world-model representations for efficient robot planning," tech. rep., Arizona State University, 2024.
[16] G. Konidaris, "On the necessity of abstraction," *Current opinion in behavioral sciences*, vol. 29, pp. 1–7, 2019.
[17] M. Dadvar, R. K. Nayyar, and S. Srivastava, "Conditional abstraction trees for sample-efficient reinforcement learning," in *Uncertainty in Artificial Intelligence*, pp. 485–495, PMLR, 2023.
[18] J. Hong, S. Levine, and A. Dragan, "Learning to influence human behavior with offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[19] K. Mahadevan, J. Chien, N. Brown, Z. Xu, C. Parada, F. Xia, A. Zeng, L. Takayama, and D. Sadigh, "Generative expressive robot behaviors using large language models," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 482–491, 2024.
[20] Y. Che, A. M. Okamura, and D. Sadigh, "Efficient and trustworthy social navigation via explicit and implicit robot–human communication," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 692–707, 2020.
[21] V. V. Unhelkar, S. Li, and J. A. Shah, "Decision-making for bidirectional communication in sequential human-robot collaborative tasks," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 329–341, 2020.
[22] T. Kaupp, A. Makarenko, and H. Durrant-Whyte, "Human–robot communication for collaborative decision making—a probabilistic approach," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 444–456, 2010.
[23] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.
[24] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
[25] F. Giunchiglia and T. Walsh, "A theory of abstraction," *Artificial intelligence*, vol. 57, no. 2-3, pp. 323–389, 1992.
[26] L. Li, T. J. Walsh, and M. L. Littman, "Towards a unified theory of state abstraction for mdps.," in *AI&M*, 2006.