# Observer-Aware Planning with Implicit and Explicit Communication

Shuwa Miura
University of Massachusetts Amherst
Amherst, USA
smiura@umass.edu

Shlomo Zilberstein
University of Massachusetts Amherst
Amherst, USA
shlomo@cs.umass.edu

## ABSTRACT

This paper presents a computational model designed for planning both implicit and explicit communication of intentions, goals, and desires. Building upon previous research focused on implicit communication of intention via actions, our model seeks to strategically influence an observer's belief using both the agent's actions and explicit messages. We show that our proposed model can be considered to be a special case of general multi-agent problems with explicit communication under certain assumptions. Since the mental state of the observer depends on histories, computing a policy for the proposed model amounts to optimizing a non-Markovian objective, which we show to be intractable in the worst case. To mitigate this challenge, we propose a technique based on splitting domain and communication actions during planning. We conclude with experimental evaluations of the proposed approach that illustrate its effectiveness.

## KEYWORDS

MDP; legibility; theory of mind; HRI

**ACM Reference Format:**
Shuwa Miura and Shlomo Zilberstein. 2024. Observer-Aware Planning with Implicit and Explicit Communication. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024,* IFAAMAS, 9 pages.

## 1 INTRODUCTION

Communication of intentions, goals, and desires is integral to our daily interactions, making them essential for autonomous agents. Communication can manifest itself in both implicit and explicit forms. Take, for instance, a scenario in which an autonomous vehicle (AV) must navigate around an obstacle on the road, as illustrated in Fig. 1. Simultaneously, another vehicle approaches from the opposite lane. The AV, or "ego vehicle," faces two primary options: it can yield to the oncoming car or assert its right of way. Implicit communication through behavior is one method the ego vehicle might use to convey its intentions. For instance, if the ego vehicle slows down or shifts towards the right edge of its lane, these actions could be interpreted as signs of yielding. Conversely, if the ego vehicle speeds up or veers towards the center or left side of the lane, it indicates a likely refusal to yield. Alternatively, the AV could employ explicit communication methods, such as flashing its headlights or using other external human-machine interfaces (eHMIs). The combination of both implicit and explicit communication forms is vital for effectively conveying intentions [6]. However, limited research has tackled the challenge of planning the execution of both domain actions and explicit communication.

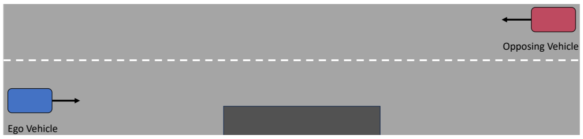


**Figure 1: AV Obstacle Avoidance**

In this paper, we present a computational model designed to strategize both implicit and explicit communication of intentions, goals, and desires. Our model builds on the Observer-Aware Markov Decision Process (OAMDP) framework [27]. OAMDP offers a unified approach for planning implicit communication through actions. This form of communication encompasses a wide range of behaviors. For example, *legible* behavior [8, 26] conveys intentions implicitly via chosen actions. In contrast, *deceptive* behavior [7, 25] either obscures or actively misrepresents an agent's intentions. *Predictable* behavior enables observers to anticipate subsequent actions [9]. Finally, agents can manifest their *capability* or *incapability* through their choice of action [23]. The OAMDP framework employs a "theory-of-mind" (ToM) approach, hypothesizing a model that represents how observers interpret the agent's actions. Here, we introduce the Communicative Observer-Aware MDP (Com-OAMDP), which extends OAMDPs with explicit communication.

To contextualize Com-OAMDP within the existing literature, we highlight that it can be seen as a special case of the Communicative Interactive POMDP (CIPOMDP) [13]. CIPOMDP extends the Interactive POMDP (IPOMDP) [14], a framework designed for general multi-agent planning that subjectively models the behaviors of other agents, with explicit communications. While (C)IPOMDP is intended for general multi-agent problems, permitting joint actions and partial observability, we show that our proposed model Com-OAMDP can be seen as a special case for multi-agent scenarios where the observer's role is assumed to be passive and the environment is fully observable (Proposition 2).

Even with the assumption of a passive observer and full observability, however, we show that computing an optimal policy for Com-OAMDPs is PSPACE-hard (Proposition 1). To address this challenge, we use Monte-Carlo Tree Search (MCTS) to solve Com-OAMDPs and propose a technique based on splitting domain and communication actions during planning. Our empirical evaluation reports the results of solving Com-OAMDPs instances using MCTS and shows the effectiveness of the proposed modifications to MCTS.

*Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.* © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

# 2 BACKGROUND

## 2.1 MDP

A Markov decision process (MDP) models sequential decision-making in environments with stochastic effects. An MDP can be characterized by the tuple $M = \langle S, A, T, R, \gamma, s_0 \rangle$. $S$ is a set of states. $A$ is a set of actions. $T(s_t, a_t, s_{t+1})$ is the probability of $S_{t+1}{=}s_{t+1}$ when $A_t{=}a_t$ and $S_t{=}s_t$. $R$ is a conditional distribution of reward given $s_t, a_t$. $\gamma$ is a parameter called the discount factor. Without loss of generality, we assume there is one initial state $s_0$. The absorbing terminal state always transitions back to itself with zero reward.

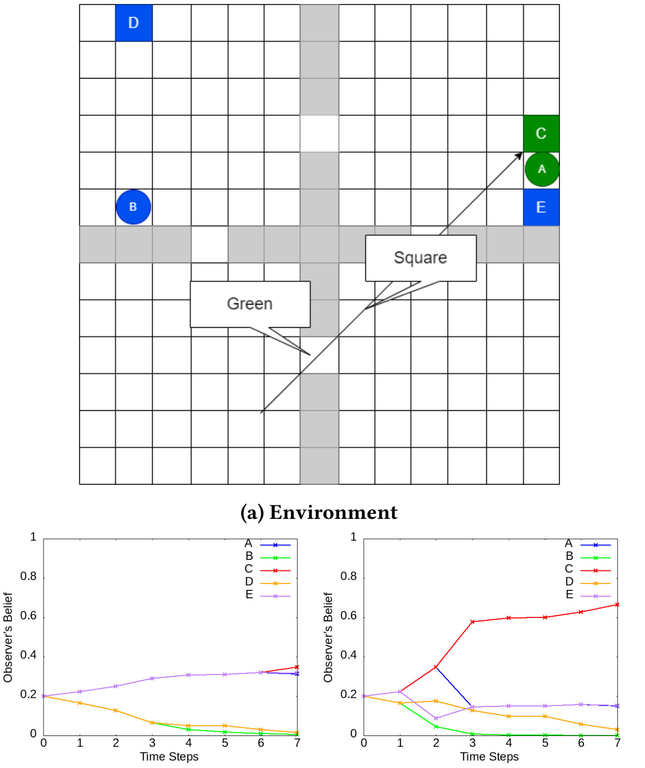
A policy, denoted as $\pi$, dictates the course of action. We use the following two types of policies in the paper. A *stationary policy* is a conditional distribution of actions given a state. When $\pi$ is deterministic, it is a mapping from $S$ to $A$. A *history-dependent policy* is a conditional distribution of actions given a history, where a history $h_{t+1}$ is a sequence of state-action pairs up to time $t$ and the last visited state $s_{t+1}$. The return of a history is the discounted sum of rewards. An optimal policy for an MDP is a policy that maximizes expected return. A policy $\pi$ induces a value function $V^\pi(s)$, which represents the expected return given a policy $\pi$ starting from state $s$. Similarly, a policy $\pi$ induces a Q-value $Q^\pi(s, a)$, which represents the expected return given a policy $\pi$ starting from state $s$ and taking action $a$. The optimal Q-value refers to the Q-value for an optimal policy.

## 2.2 Observer-Aware MDP

Observer-Aware Markov Decision Process (OAMDP) extends the MDP by allowing the reward to depend on the observer's assumed belief about the type of observed agent [27]. For example, in the AV example earlier, the possible types are yielding and insisting its way. After the ego vehicle decelerates, for example, the observer updates its belief about the intention of the ego vehicle. Formally,

**Definition.** An OAMDP is a tuple
$M = \langle S, A, T, \gamma, s_0, \Theta, B, R \rangle$ where:

- $S$ is a finite set of states.
- $A$ is a finite set of actions.
- $T$ describes the transition probability.
- $\gamma$ is a discount factor.
- $s_0$ is the initial state.
- $\Theta$ is a finite set of *types*, representing a characteristic of the agent such as possible goals, intentions, or capabilities. Types in OAMDPs are analogous to types in Bayesian games [17].
- $B : H^* \longrightarrow \Delta^{|\Theta|}$ represents the assumed belief of the observer given a history. $H^*$ is the set of all finite histories and $\Delta^{|\Theta|}$ is a simplex on $\Theta$.
- $R : S \times A \times \Delta^{|\Theta|} \longrightarrow \mathbb{R}$ describes how desirable it is to take an action given a state and a belief $b \in \Delta^{|\Theta|}$. We assume, in this paper, that the rewards can be represented as a convex combination of *domain* and *belief-dependent* rewards. That is, $R(s_t \in S, a_t \in A, b_t \in \Delta^{|\Theta|}) = \lambda R_d(s_t, a_t) + (1 - \lambda) R_b(b_t)$ for $\lambda \in [0, 1]$, where $R_d$ and $R_b$ represent domain and belief-dependent reward, respectively. Note that the reward depends on histories through the beliefs.



**(a) Environment**

**(b) Without messages**

**(c) With messages**

**Figure 2: Example of combining implicit and explicit communication in the Maze World environment ($\beta = 0.3$).**

A solution to OAMDPs is a policy that maximizes the expected discounted return for a given horizon $K$:

$$\mathbb{E}[\sum_{t=0}^{K} \gamma^t R(S_t, A_t, B(H_t)) | S_0 = s_0, \pi]. \tag{1}$$

Intuitively, OAMDP is an extension of MDP, which assumes how the observer would interpret the agent's behavior ($B$) and what interpretation is more desirable ($R$). OAMDPs can produce various observer-aware behaviors by changing $R$. For example, if the goal is being legible [8], $R$ could be the negative total variation (TV) or Euclidean distance between the current belief and the target belief ($b(\theta^*) = 1$ for the true type $\theta^* \in \Theta$). On the other hand, if the ego agent wants to obscure its intention, rewards could be the entropy of the observer's belief.

**Bayesian Observer** While the definition of general OAMDP allows for any function to serve as $B$, we will henceforth assume that the observer is Bayesian. The observer's belief is updated using:

$$b_{t+1}(\theta | h_{t+1}) \propto \hat{\Pr}(s_{t+1}, a_t | s_t, \theta) b_t(\theta | h_t) \tag{2}$$

Here, $\hat{\Pr}(s_{t+1}, a_t | s_t, \theta)$ represents the probability, according to the observer's model, that the observed agent would perform action $a_t$ and subsequently arrive at state $s_{t+1}$ given $s_t$ and the type $\theta$. We denote this probability with $\hat{\Pr}$ instead of Pr to indicate that this is the observer's assumed model.

**Noisy Rational Model** In modeling the observer, a common strategy is to use inverse planning under the assumption that the observed agent exhibits approximately rational behavior given its intention. Baker et al. [3] explored the connection between Bayesian reasoning and human understanding of goals. A model presented in their work presumes noisy rationality:

$$\pi_\theta(s_t, a_t) \propto \exp^{\beta Q^*_\theta(s_t,a_t)} \tag{3}$$

In the equation above, $Q^*_\theta(s_t, a_t)$ denotes the optimal Q-value given the type $\theta$. $\beta \in \mathbb{R}$ serves as a hyper-parameter representing the agent's rationality level. Intuitively, it is assumed that the observed agent selects an action at a state with a probability exponentially proportional to the quality of the action at the current state.

Subsequently, upon observing $h_{t+1}$, the observer updates the posterior belief over goals (or types) according to:

$$b_{t+1}(\theta|h_{t+1}) = \frac{T_\theta(s_t, a_t, s_{t+1})\pi_\theta(s_t, a_t)b_t(\theta|h_t)}{\sum_{\theta' \in \Theta} T_{\theta'}(s_t, a_t, s_{t+1})\pi_{\theta'}(s_t, a_t)b_t(\theta'|h_t)}. \tag{4}$$

In the equation above, $\theta \in \Theta$ represents the agent's type, $b_t$ signifies the observer's prior belief for each $\theta$, $T_\theta$ is the transition function and $\pi_\theta$ is the presupposed policy given a type $\theta$. When $T_\theta$ is the same for all $\theta \in \Theta$, Equation 4 simplifies to:

$$b_{t+1}(\theta|h_{t+1}) = \frac{\pi_\theta(s_t, a_t)b_t(\theta|h_t)}{\sum_{\theta' \in \Theta} \pi_{\theta'}(s_t, a_t)b_t(\theta'|h_t)}. \tag{5}$$

When the action is not observable to the observing agent, the belief can be updated by marginalizing over actions:

$$b_{t+1}(\theta|h_{t+1}) = \frac{\sum_{a_t \in A} T_\theta(s_t, a_t, s_{t+1})\pi_\theta(s_t, a_t)b_t(\theta|h_t)}{\sum_{\theta' \in \Theta} \sum_{a_t \in A} T_{\theta'}(s_t, a_t, s_{t+1})\pi_{\theta'}(s_t, a_t)b_t(\theta'|h_t)}. \tag{6}$$

For instance, Fig. 2 illustrates a Maze World from [3] where an agent can choose from nine different actions: *Stay, North, South, East, West, NorthEast, NorthWest, SouthEast,* and *SouthWest.* Notably, when the agent takes action, it can deviate to the left or right—akin to veering off its chosen direction—with a 0.05 probability each. The agent's objective is to reach one of the potential goals: *A*, *B*, *C*, *D*, or *E*. We assign five possible types to the agent, each corresponding to a particular goal ($\Theta = \{\theta_A, \theta_B, \theta_C, \theta_D, \theta_E\}$). The rewards are proportional to the negative of the distance traveled.

In Fig 2a, the agent moves *NorthEast* in the initial timestep. Since this action is not optimal for goal *B* and *D*, the Q-values $Q^*_{\theta_B}(s_0, \mathit{NorthEast})$ and $Q^*_{\theta_D}(s_0, \mathit{NorthEast})$ are lower compared to alternative actions like *NorthWest.* According to the noisy rational model (Equation 3), the likelihood of the observed agent taking the *NorthEast* action is lower than *NorthWest* given that the goal is *B* or *D*. That is, $\pi_{\theta_B}(s_0, \mathit{NorthEast}) < \pi_{\theta_B}(s_0, \mathit{NorthWest})$ and $\pi_{\theta_D}(s_0, \mathit{NorthEast}) < \pi_{\theta_D}(s_0, \mathit{NorthWest})$. Conversely, moving *NorthEast* is the optimal choice for goals *A*, *C* or *E*. As per the noisy rational model (Equation 3), the probability of the observed agent taking the *NorthEast* action exceeds that of taking other sub-optimal actions.

Fig. 2b visualizes the observer's belief changes according to Equation 6. The plot shows that the belief in $\theta_B$ and $\theta_D$ diminishes, while beliefs in $\theta_A$, $\theta_C$, and $\theta_E$ rise after the agent's initial *NorthEast* move. Since the agent keeps moving *NorthEast*, the observer's belief on *A*, *C*, and *E* remain equally high.

**Relationship to POMDP and IPOMDP** Despite the pronounced similarities between OAMDPs and POMDPs [19], one does not subsume the other. Both formulations operate based on the agents' beliefs. Yet, a crucial distinction exists: POMDP focuses on the acting agent's beliefs regarding states, whereas OAMDP centers on the observer's presumed beliefs about the types of the observed agent. Additionally, while rewards in POMDPs are defined by the underlying states, rewards in OAMDPs depend on the observers' beliefs.

OAMDP can be seen as a special case of Interactive POMDP (IPOMDP). IPOMDP is a model for general multi-agent scenarios based on subjectively modeling behaviors of other agents. In principle, observer-aware planning problems can be framed as general multi-agent problems such as IPOMDP. However, solving IPOMDP is notoriously difficult [30]. Prior work [27] showed that OAMDP could be derived from IPOMDP, given certain additional assumptions.

**Solving OAMDPs** We next describe solution methods for OAMDPs. Traditional solution methods for MDPs, such as Value Iteration, are not directly applicable to OAMDPs. This is primarily due to their reliance on the Markov property of rewards, which is a property not possessed by OAMDPs. Previous work [27] proposed solving OAMDPs using Monte-Carlo Tree Search (MCTS), where each node in the search tree represents a pair consisting of a domain state and an observer belief.

Computing an optimal policy for OAMDPs with a Bayesian observer, however, is shown to be intractable in the worst case [27]. Since complexity classes are defined in relation to decision problems, the result is formally shown for the finite-horizon value problem: given an OAMDP with a Bayesian observer, a planning horizon $K$, and a threshold $Th$, does the OAMDP possess a (finite-horizon history-dependent) policy with value equal to or exceeding $Th$? The finite-horizon value problem for OAMDPs is shown to be PSPACE-hard via reduction from QSAT [27].

## 3 PLANNING IMPLICIT AND EXPLICIT COMMUNICATION

In this section, we detail how to model explicit communication alongside implicit communication, in the context of OAMDP. In addition to the existing set of domain or task actions, $A$, we introduce a set of communication actions, $\mathbb{M}$. At each time step, the agent performs a pair of actions: one from the task execution and one for communication ($A \times \mathbb{M}$). The observer is presumed to update its belief based on $\hat{\Pr}(a_t, m_t, s_{t+1}|s_t, \theta)$, where $m_t$ signifies the message sent at time $t$.

**Definition.** A Communicative OAMDP (Com-OAMDP) is a special case of OAMDP defined by the tuple $M = \langle S, A \times \mathbb{M}, T, \gamma, s_0, \Theta, B, R\rangle$ with the following specifications:

- $\mathbb{M}$ denotes a set of messages.
- $T$ does not depend on the messages $m_t \in \mathbb{M}$.

It is important to note that histories in Com-OAMDPs incorporate messages along with domain actions ($h_t = s_0 a_0 m_0 s_1 \cdots a_{t-1} m_{t-1} s_t$). The function $B$ maps these histories, which include messages, to the observer's beliefs. Similarly, the function $R_d : S \times A \times \mathbb{M} \rightarrow \mathbb{R}$ also depends on messages. The set $\mathbb{M}$ may include a *nil* message, which implies no message is being sent.

For instance, in Fig. 2, one possible set of available messages is $\mathbb{M} = \{blue, green, square, circle, nil\}$. Sending the message *green* signals to the observer that the observed agent intends to reach one of the green goals ($A$, $C$, or $D$).

As in the OAMDP framework, we proceed in this paper by considering a scenario where the observer is Bayesian and updates its belief as follows:

$$b_{t+1}(\theta|h_{t+1}) \propto \hat{\Pr}(a_t, m_t, s_{t+1}|s_t, \theta) b_t(\theta|h_t) \tag{7}$$

Note that this belief update incorporates both a domain action $a_t$ and a message $m_t$.

The updated belief can be expressed as:

$$b_{t+1}(\theta|h_{t+1}) = \frac{T_\theta(s_t, a_t, s_{t+1}) \hat{\Pr}(a_t, m_t|s_t, \theta) b_t(\theta|h_t)}{\sum_{\theta' \in \Theta} T_{\theta'}(s_t, a_t, s_{t+1}) \hat{\Pr}(a_t, m_t|s_t, \theta') b_t(\theta'|h_t)}. \tag{8}$$

We discuss one possible way define the semantics of messages ($\hat{\Pr}(a_t, m_t|s_t, \theta)$) next.

**Generative Noise Model** One potential way to model the probability of sending a message posits that the observed agent sends a correct message with a certain probability, $0 \leq \alpha \leq 1$, sends an incorrect message with $0 \leq \epsilon \leq 1$ and sends out a *nil* message otherwise, similar to the model proposed in [13]. This model interprets each message as a unary predicate on types. For instance, the message *green* equates to $green(\theta_A) = green(\theta_C) = green(\theta_D) = 1$ and $green(\theta_B) = green(\theta_D) = 0$. Formally, the probability of transmitting a message is:

$$\hat{\Pr}(m_t|s_t, \theta) = \begin{cases} \alpha \frac{1}{|\{m|m \in \mathbb{M}, m(\theta)=1\}|} & m_t(\theta) = 1 \\ \begin{cases} 1 & \text{if no } m(\theta) = 1 \text{ exits} \\ 1 - \alpha - \epsilon & \text{otherwise} \end{cases} & m_t = nil \\ \epsilon \frac{1}{|\{m|m \in \mathbb{M}, m(\theta)=0\}|} & \text{otherwise} \end{cases}$$

As an example, $\Pr(green|\cdot, \theta_C) = \Pr(square|\cdot, \theta_C) = \frac{1}{2}\alpha$, $\Pr(blue|\cdot, \theta_C) = \Pr(circle|\cdot, \theta_C) = \frac{1}{2}\epsilon$, and $\Pr(nil|\cdot, \theta_C) = 1 - \alpha - \epsilon$. In this paper, we combine the generative noise model with the noisy rational model, and assumes that $\hat{\Pr}(a_t, m_t|s_t, \theta) = \hat{\Pr}(a_t|s_t, \theta)\hat{\Pr}(m_t|s_t, \theta)$.

Fig. 2c illustrates the observer's belief changes corresponding to the path in Fig. 2a. As the agent moves away from the goal $B$ and $D$, the belief in $B$ and $D$ decreases over time, but the observer cannot differentiate between $A$, $C$ and $E$ with information emitted from domain actions alone. Yet, when the agent sends the *green* message, the belief on $E$ decreases, given that heading to $E$ is in contradiction to the conveyed message. When the agent sends out the message *square* next, the belief on $A$ gets lowered, allowing the observer to confidently infer that the agent is heading towards $C$.

Note that the agent could have sent the message *green* immediately in the example. Although this action would instantly diminish the belief in $B$ and $E$, the observer would have high beleif in $A$, $C$, and $D$. To prevent this scenario, the agent depicted in Fig. 2a waits until the belief in $D$ is sufficiently reduced. This example emphasizes the advantage of considering both implicit and explicit communication.

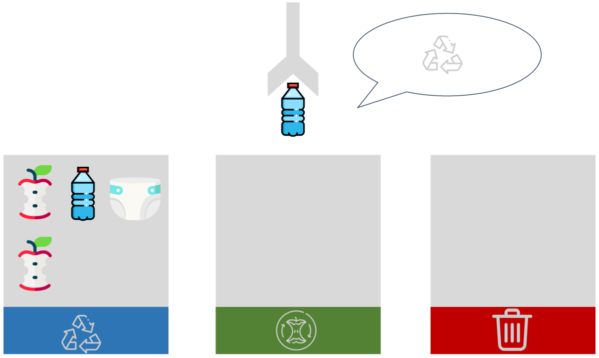


**Figure 3: Recycle problem**

It is important to note that the general definition of Com-OAMDP is not tied to the noisy rational model or the generative noise model. Instead, Com-OAMDPs using these models represent specific instances within the broader Com-OAMDP framework. Com-OAMDP serves as a comprehensive framework addressing scenarios where rewards depend on the observer's beliefs, updated via both implicit and explicit communication.

## 4 EXAMPLES

In this section, we describe a few examples of Com-OAMDPs.

**Recycling** In the recycling problem depicted in Fig. 3, a robot arm is tasked with sorting garbage. The garbage comprises three types of items: food waste, diapers, and water bottles. Each type must be placed in its designated bin: food waste in compost, diapers in trash, and water bottles in recycle. The observer is aware that the robot is programmed to dispose of food waste in compost. However, it remains uncertain whether the robot correctly identifies that diapers belong in the trash rather than compost, or whether water bottles should be recycled instead of being thrown in the trash. Thus, there are four possible types for the observed agent. At every time step, the robot arm can choose to either pick up an item or attempt to deposit it into one of the bins. However, the robot arm's actions are not flawless. It only successfully places an item in the intended bin with a probability $p$. When an observer sees an incorrectly placed item, it's unclear whether the robot made an action error or misclassified the item. To address this confusion, the robot is equipped with a communication feature, allowing it to indicate the intended bin for the item it holds. However, sending a message incurs a cost. The robot needs to plan its course of action, depending on the cost of communication and initial configurations of objects.

**Obstacle Avoidance** is an AV example introduced earlier (Fig. 1). There is an obstacle in front of the ego vehicle. To avoid the obstacle, the ego vehicle needs to go to the left side of the lane. However, a vehicle approaches from the opposite direction. The possible intentions for the ego vehicles are insisting its way or yielding for the opposite car. The ego vehicle's state is defined by its speed $0 \leq \phi \leq 30$, velocity $0 \leq \dot{\phi} \leq 4$, and its position in the lane (left, center, or right). The available actions involve changing lanes and adjusting velocity by increments of −2,−1,0,1, and 2. Additionally, the ego vehicle can flash its headlights to communicate its intention explicitly. However, there is some cost associated with sending a message. The ego vehicle needs to plan its course of action depending on the cost of communication.

## 5 THEORETICAL PROPERTIES

**Complexity of Com-OAMDP** Similarly to OAMDP, the generality of the Com-OAMDP, however, does come with the drawback of potential intractability in the worst case. Fundamentally, as we allow the reward to depend on the observer's belief, which in turn relies on the history up to that point, policy computations for Com-OAMDPs are prone to the curse of history.

**Proposition 1.** The finite-horizon value problem for Com-OAMDPs with Bayesian observer is PSPACE-complete.

Proof. The complexity of the OAMDP has been proven to be PSPACE-hard through the reduction from QSAT [27]. The OAMDP in the reduction can be interpreted as a Com-OAMDP with only the *nil* message available. □

This statement suggests that finding optimal policies for Com-OAMDPs may be computationally challenging in the worst case. We will suggest a potential approach to alleviate this issue later in this paper.

### 5.1 Com-OAMDP as a Subclass of CIPOMDP

In connecting Com-OAMDP to prior work in the literature, we demonstrate that Com-OAMDP can be derived from CIPOMDP [13] given a set of additional assumptions. CIPOMDP is an extension of IPOMDP that includes explicit communication. In this section, we illustrate that Com-OAMDP is a special case of generalized multi-agent problems that involve explicit communication. This finding parallels how OAMDP can be derived from IPOMDP under a specific set of assumptions.

**IPOMDP** We first introduce IPOMDP. We assume without loss of generality that there are two agents (agent $i$ and $j$). A finitely-nested IPOMDP for agent $i$ is defined by the 6-tuple $\langle IS^{i,l}, A^i, \Omega^i, T^i, O^i, R^i \rangle$ where $IS^{i,l}$ is a set of interactive states. Interactive states consist of a domain state as well as a model of the other agent $j$. Formally, interactive states are recursively defined as follows:

$$\begin{aligned}
&IS^{i,0} = S && \Theta^{j,0} = \{\langle b^{j,0}, \hat{\theta}_j \rangle : b^{j,0} \in \Delta^{|S|}\} \\
&IS^{i,1} = S \times \Theta^{j,0} && \Theta^{j,1} = \{\langle b^{j,1}, \hat{\theta}_j \rangle : b^{j,1} \in \Delta^{|IS^{j,1}|}\} \\
&IS^{i,2} = S \times \Theta^{j,0} \times \Theta^{j,1} && \Theta^{j,2} = \{\langle b^{j,2}, \hat{\theta}_j \rangle : b^{j,2} \in \Delta^{|IS^{j,2}|}\} \\
&\cdots && \\
&IS^{i,l} = S \times_{k=0}^{l-1} \Theta^{j,k} && \Theta^{j,l} = \{\langle b^{j,l}, \hat{\theta}_j \rangle : b^{j,l} \in \Delta^{|IS^{j,l}|}\}
\end{aligned}$$

where $\hat{\theta}_j = \langle A_j, \Omega_j, T_j, O_j, R_j \rangle$ is called *frame* of agent $j$ representing all components of $j$'s model except for the belief. At each strategy level $l$, agent $i$ assumes that agent $j$ chooses an action $a^j$ rationally based on its model $\theta^{j,l-1}$, which in turn model agent $i$ at level $l-2$ and so on. IPOMDP can be seen as POMDP defined over interactive states, and optimal policies and all the other components ($A^i$, $\Omega^i$, $T^i$, and $R^i$) are defined as in POMDP. Note that the transition function depends on both $a^i$ and $a^j$.

With these definitions in place, we can now write down the belief update equation for IPOMDP [14]:

$$\begin{aligned}
b^i_{t+1}(is^i_{t+1}) = \Pr(is^i_{t+1} | a^i_t, \omega^i_{t+1}, b^i_t) = \eta \sum_{is^i_t} b^i_t(is_t) \\
\sum_{a^j_t \in A^j} \Pr(a^j_t | \theta^j_t) O^i(s_{t+1}, a_t, \omega^i_{t+1}) \Pr(is^i_{t+1} | is^i_t, a_t)
\end{aligned} \tag{9}$$

where $is^i_t$ ranges over interactive states sharing the frame with $is^i_{t+1}$, $\eta$ is a normalizing constant, and $\Pr(is^i_{t+1} | is^i_t, a_t)$ is the transition probability between interactive states:

$$\begin{aligned}
&Pr(is^i_{t+1} | is^i_t, a_t) \\
&= T^i(s_t, a^i_t, a^j_t, s_{t+1}) \sum_{\omega^j_{t+1} \in \Omega^j} O^j(a^i_t, a^j_t, s^j_{t+1}, \omega^j_{t+1}) \tau_{\hat{\theta}^j}(b^j_t, a^j_t, \omega^j_{t+1}, b^j_{t+1})
\end{aligned} \tag{10}$$

where $\tau_{\hat{\theta}^j}(b^j_t, a^j_t, \omega^j_{t+1}, b^j_{t+1})$ is 1 when $b^j_{t+1}$ equals the result of updating $b^j_t$ given $a^j_t$ and $\omega^j_{t+1}$ according to $\hat{\theta}^j$.

**CIPOMDP** CIPOMDP is an extension of IPOMDP with additional communication actions. A finitely-nested CIPOMDP for agent $i$ is defined by the 7-tuple $\langle IS^{i,l}, A^i, \mathbb{M}^i, \Omega^i, T^i, O^i, R^i \rangle$ where $\mathbb{M}^i$ is a set of messages agent $i$ can send and receive. All the other elements are defined analogously to IPOMDP except that $R^i$ now also depends on messages.

The belief update for CIPOMDP is derived as follows [13, 20][1]:

$$\begin{aligned}
&b^i_{t+1}(is^i_{t+1}) = \Pr(is^i_{t+1} | a^i_t, m^{i\rightarrow j}_t, \omega^i_{t+1}, m^{i\leftarrow j}_{t+1}, b^i_t) \\
&= \alpha \sum_{is^i_t} b^i_t(is^i_t) \sum_{a^j_t \in A^j} \Pr(m^{j\rightarrow i}_t, a^j_t | \theta^j_t) \\
&\times O^i(s_{t+1}, a_t, \omega^i_{t+1}) \Pr(is^i_{t+1} | is^i_t, a_t, m^{i\rightarrow j}_t, m^{i\leftarrow j}_{t+1})
\end{aligned} \tag{11}$$

where $is^i_t$ ranges over interactive states sharing the frame with $is^i_{t+1}$, $\alpha$ is a normalizing constant, $m^{i\rightarrow j}_t$ is the message sent by agent $i$ to $j$ at time $t$, $m^{i\leftarrow j}_{t+1}$ is the message agent $i$ received from $j$ at time $t+1$. Note that since messages are perfectly transmitted, $m^{i\rightarrow j}_t = m^{j\leftarrow i}_{t+1}$ and $m^{i\leftarrow j}_{t+1} = m^{j\rightarrow i}_t$.

$\Pr(is^i_{t+1} | is^i_t, a_t, m^{i\rightarrow j}_t, m^{i\leftarrow j}_{t+1})$ is the transition probability between interactive states:

$$\begin{aligned}
&\Pr(is^i_{t+1} | is^i_t, a_t, m^{i\rightarrow j}_t, m^{i\leftarrow j}_{t+1}) = T^i(s_t, a_t, s_{t+1}) \times \\
&\sum_{\omega^j_{t+1} \in \Omega^j} O^j(s_{t+1}, a_t, \omega^i_{t+1}) \tau_{\hat{\theta}^j}(b^j_t, a^j_t, m^{j\rightarrow i}_t, \omega^j_{t+1}, m^{j\leftarrow i}_{t+1}, b^j_{t+1})
\end{aligned} \tag{12}$$

where $\tau_{\hat{\theta}^j}(b^j_t, a^j_t, m^{j\rightarrow i}_t, \omega^j_{t+1}, m^{j\leftarrow i}_{t+1}, b^j_{t+1})$ is 1 when $b^j_{t+1}$ equals the result of updating $b^j_t$ given $a^j_t$, $m^{j\rightarrow i}_t$, $\omega^j_{t+1}$, $m^{j\leftarrow i}_{t+1}$ according to $\hat{\theta}^j$.

**Com-OAMDP Assumptions** We now show that Com-OAMDPs form a special case of CIPOMDPs under the following four assumptions:

A1 The observing agent is passive. Formally, we can represent this assumption by having only one action for the observing agent $A^j = \{noop\}$ and assuming that the observing agent always sends *nil* message.

A2 The type of the observing agent is initially known to the observed agent.

[1] The update assumes that messages are transmitted perfectly without errors.

A3 The observing agent can fully observe the underlying states ($S$) and actions performed by the observed agent ($A^i$), i.e. $\Omega^j = S \times A^i$ and $O^j(a^i_t, a^j_t, s_{t+1}, \omega^j_{t+1}) = 1$ if $\omega^j_{t+1} = \langle s_{t+1}, a^i_t \rangle$, and 0 otherwise.

A4 The observed agent can fully observe the underlying states ($S$), i.e. $\Omega^i = S$ and $O^i(a^i_t, a^j_t, s_{t+1}, \omega^j_{t+1}) = 1$ if $\omega^j_{t+1} = s_{t+1}$, and 0 otherwise.

**Proposition 2.** Under Assumptions A1-4, for a given instance of a CIPOMDP, there exists an equivalent instance of Com-OAMDP.

Proof Sketch. With Assumption A1 and A3, the transition function between interactive states simplifies to:

$$
\begin{aligned}
&\Pr(is^i_{t+1} | is^i_t, a_t, m^{i\to j}_t, m^{i\leftarrow j}_{t+1}) = T^i(s_t, a_t, s_{t+1}) \times \\
&\sum_{\omega^j_{t+1} \in \Omega^j} O^j(s_{t+1}, a_t, \omega^j_{t+1}) \tau_{\hat{\theta}^j}(b^j_t, a^j_t, m^{j\to i}_t, \omega^j_{t+1}, m^{j\leftarrow i}_{t+1}, b^j_{t+1}) \\
&= T^i(s_t, a_t, s_{t+1}) \tau_{\hat{\theta}^j}(b^j_t, a^j_t, m^{j\to i}_t, a^i_t, s_{t+1}, m^{j\leftarrow i}_{t+1}, b^j_{t+1}) \text{ by A3} \\
&= T^i(s_t, a_t, s_{t+1}) \tau_{\hat{\theta}^j}(b^j_t, a^i_t, s_{t+1}, m^{j\leftarrow i}_{t+1}, b^j_{t+1}) \text{ by A1}
\end{aligned}
\tag{13}
$$

If we define a Com-OAMDP instance with $T = T^i$ and the belief function $B$ as the belief update corresponding to $\hat{\theta}^j$, we see that the transitions in the Com-OAMDP correspond to the transitions in the original CIPOMDP.

We next show that under Assumptions A1-4, there is exactly one interactive state with $b^i_t(is^i_t) = 1$ for each timestep $t$. This would imply that the Com-OAMDP described above is equivalent to the original CIPOMDP. We show the claim by induction on the timesteps. By Assumption A2, the claim is true for the first timestep. Suppose now that at time $t$, there is exactly one $is^i_* = \langle s_t, \langle b^j_*, \hat{\theta}^j{}_* \rangle \rangle$ such that $b_t(is^i_*) = 1$, then:

$$
\begin{aligned}
&b^i_{t+1}(is^i_{t+1}) = \Pr(is^i_{t+1} | a^i_t, m^{i\to j}_t, \omega^i_{t+1}, m^{i\leftarrow j}_{t+1}, b^i_t) \\
&= \alpha \sum_{is^i_t} b^i_t(is^i_t) \sum_{a^j_t \in A^j} \Pr(m^{j\to i}_t, a^j_t | \theta^j_t) O^i(s_{t+1}, a_t, \omega^i_{t+1}) \times \\
&\Pr(is^i_{t+1} | is^i_t, a_t, m^{i\to j}_t, m^{j\to i}_{t+1}) \\
&= \alpha \sum_{is^i_t} b^i_t(is^i_t) \delta(\omega^i_{t+1}, s_{t+1}) \Pr(is^i_{t+1} | is^i_t, a_t, m^{i\to j}_t, m^{j\to i}_{t+1}) \text{ by A1,4} \\
&= \alpha \delta(\omega^i_{t+1}, s_{t+1}) \Pr(is^i_{t+1} | is^i_*, a_t, m^{i\to j}_t, m^{j\to i}_{t+1})
\end{aligned}
\tag{14}
$$

Note that the right-hand size of the equation is only positive for the one interactive state with $s_{t+1} = \omega^i_{t+1}$, where $b^j_{t+1}$ is the result of updating $b^j_*$ according to $\hat{\theta}^j_*$, and $\hat{\theta}^j = \theta^j_*$. Therefore, there is exactly one interactive state possible for time $t + 1$ as well. □

In conclusion, Com-OAMDP can be seen as a special case of CIPOMDP. Note that, the definition of Com-OAMDP with an arbitrary belief function $B$ can represent the recursive belief update $\tau_{\hat{\theta}^j}$ in CIPOMDP in principle. However, the belief update in Equation 8 uses a limited form of recursive reasoning. It assumes that the observed-agent is at the strategy level 2, who reasons about the observer at level 1, who in turn reasons about the observed-agent at level 0.

## 6 TOWARDS EFFICIENT PLANNING FOR COM-OAMDPS

In this section, we propose a technique to solve Com-OAMDPs more efficiently. As Com-OAMDP is a special case of OAMDP, MCTS can also be utilized to solve Com-OAMDPs. However, directly applying MCTS to Com-OAMDPs without accounting for its unique structure might not be optimal. Therefore, we propose a modification to MCTS specifically suited for solving Com-OAMDPs in this section.

MCTS operates through a series of stochastic simulations, starting from the root node. Each iteration of MCTS consists of four steps: (1) selection: from the root node, a child node is selected until a leaf node is reached; (2) expansion: child nodes are appended to the selected node; (3) simulation: an accumulated discounted reward is sampled by simulating a rollout policy $\pi$; (4) backpropagation: the value estimates are updated for ancestor nodes. When applied to planning problems with uncertainty, nodes in the search tree consists of *decision* and *chance* nodes. For chance nodes, child nodes are selected by simulating the action in the environment. As for decision nodes, UCT is an instance of MCTS, choosing actions according to the UCB1 formula [1]:

$$
Q(s, a) + C\sqrt{\log N(s)/N(s, a)} \tag{15}
$$

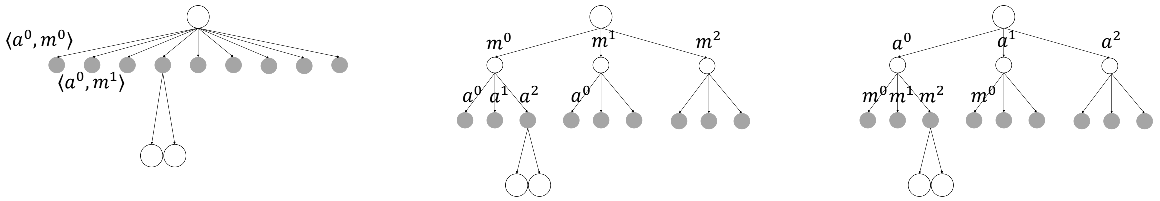
where $Q(s, a)$ is the estimated $Q$-value, $N(s)$ and $N(s, a)$ are counters of the number of times the simulations encountered the node $s$ and $(s, a)$, and $C$ is a constant that controls the degree of exploration, respectively. Note that although UCT eventually expands the whole graph and finds the optimal policy, policies returned after a fixed number of iterations are not guaranteed to be optimal.

When MCTS is applied to solve Com-OAMDPs, each chance node corresponds to a pair of physical state and the belief of the observer ($\langle s, b \rangle \in S \times \Delta^{|\Theta|}$). Each decision node corresponds to a pair of domain action and message ($\langle a, m \rangle \in A \times \mathbb{M}$).

**Rollout Policy** We used $\pi^*_d$, which selects an optimal domain action in terms of the domain rewards for the true goal. $\pi^*_d$ sends no messages during rollouts. Note that $\pi^*_d$ might not be available for all problems before planning. However, using Equation 3 requires computing $\pi^*_d$ for each possible type $\theta$.

**Move Splitting** One possible way to exploit the structure of Com-OAMDPs is splitting domain and communication actions in MCTS. Recall that in Com-OAMDPs, an action consists of a pair of domain and communication actions. Consequently, a direct implementation of MCTS would produce $|A| \times |\mathbb{M}|$ chance nodes for every decision node, as depicted in Fig. 4a. UCT, in this setup, is obligated to explore each action at least once. However, a significant portion of these chance nodes might not be worth exploring. For instance, consider applying UCT to solve the MazeWorld example we discussed earlier (Fig. 2). Transmitting an incorrect message (e.g., *blue*) is invariably a poor option, regardless of the domain action. Nevertheless, UCT must try every combination of sending the message with the domain action.

To mitigate the issue, we propose to split domain and communication actions in the search tree (Fig. 4b). We outline our modified version of UCT, termed UCT (MA), in Algorithm 1. UCT (MA) creates intermediate decision nodes for each message $m \in \mathbb{M}$ (*message node*) whenever a decision node corresponding to a belief state



**(a) vanilla MCTS** **(b) MCTS (MA)** **(c) MCTS (AM)**

**Figure 4: Search trees with and without splitting. White and gray circles represent decision and chance nodes, respectively.**

**Algorithm 1** UCT (MA)

```
1:  function UCT(rootNode)
2:      while within computational budget do
3:          $node \leftarrow SELECT(rootNode)$
4:          $child \leftarrow EXPAND(node)$
5:          $G \leftarrow SIMULATE(child)$
6:          $BACKPROPAGATE(node, G)$
7:      end while
8:      return BestAction(rootNode)
9:  end function
10:
11: function Select(node)
12:     while node is fully expanded do
13:         if node is a state node then
14:             node ← choose a message node according to UCB1
15:         else if node is a message node then
16:             node ← choose an action node according to UCB1
17:             node ← choose a state node according to $T$
18:         end if
19:     end while
20:     return node
21: end function
22:
23: function BACKPROPAGATE(node, G)
24:     if node is a state node then
25:         $N(s) += 1$
26:         return $BACKPROPAGATE(node.grandParent, G)$
27:     else if node is a message node then
28:         $N(s, m) += 1$
29:         $N(s, \langle a, m \rangle) += 1$
30:         $G \leftarrow R(s, \langle a, m \rangle, m) + \gamma G$
31:         $Q(s, m) \leftarrow Q(s, m) + \frac{1}{N(s,m)}(G - Q(s, m))$
32:         $Q(s, \langle a, m \rangle) \leftarrow Q(s, \langle a, m \rangle) + \frac{1}{N(s,\langle a,m \rangle)}(G - Q(s, \langle a, m \rangle))$
33:         return $BACKPROPAGATE(node.parent, G)$
34:     end if
35: end function
```

(*state node*) is selected for expansion. When a message node corresponding to message $m$ is selected for expansion, chance nodes are created for each domain action $a \in A$ (*action nodes*).

In the selection phase, both state and message nodes choose child nodes based on the UCB1 formula, as detailed in Equation 15 (see line 14 and line 16). At action nodes, child nodes are selected according to simulating the next state in the environment (line 17). Each state node maintains the visitation count ($N(s)$). Each message node maintains the visitation count $N(s, m)$ as well as the estimate $Q(s, m)$, which represents the estimated value given the transmission of message $m$. Similarly, each action node maintains the visitation count $N(s, \langle a, m \rangle)$ and the estimate $Q(s, \langle a, m \rangle)$. The statistics are updating with Monte-Carlo update (line 23-35).

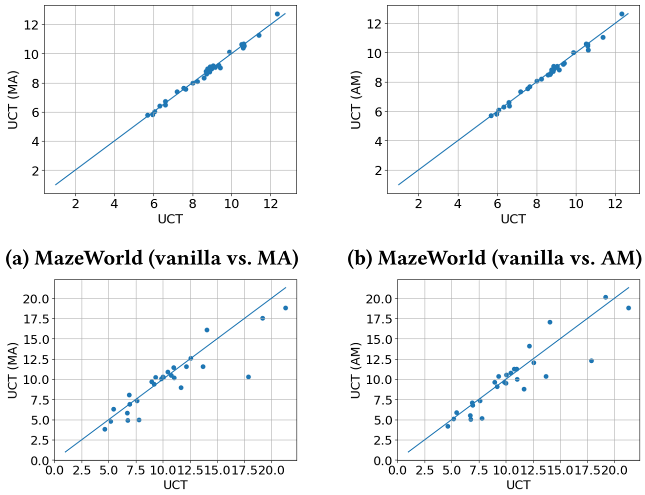
In UCT (MA), both state and message nodes can execute a simulation using a rollout policy. A rollout policy is well-defined for state nodes, but not for message nodes. When simulating from a message node, a domain action suggested by a rollout policy is combined with the corresponding message for one step. Then a simulation is performed using only domain actions. With this action-splitting strategy, UCT (MA) is expected to spend less time exploring the transmission of clearly unpromising messages. This is because, once the value estimate for a node associated with an unpromising message is low, UCT (MA) will more frequently explore other, potentially more promising messages.

As illustrated in Fig. 4c, we present an alternative version of the algorithm where action nodes precede message nodes. While UCT (AM) largely mirrors UCT (MA), it differentiates by interchanging the roles of message and action nodes.

## 7 EXPERIMENTS

In this section, we present the empirical evaluation of our proposed algorithms for solving Com-OAMDP instances. We compare the performance of UCT (MA) and UCT (AM) with that of vanilla UCT. For empirical evaluation, we use random instances of the MazeWorld and Recycle problems. Each problem instance is solved online, using 10000 MCTS iterations per timestep. We test each configuration 30 times and report the average results. In our experiments, we set $C = 1.0$ for the exploration constant in the UCB1 formula. We used $\beta = 0.3$ for the constant in the noisy rational model, $\alpha = 0.4$, $\epsilon = 0.1$ in our communication model. The horizon $K$ was set to 50 for all problems. The rollout policy was run for 20 steps. The results are reported in terms of costs or negative rewards.

For MazeWorld, we created 60 random instances for the environment shown in Fig. 2. For each problem instance, five locations



**(a) MazeWorld (vanilla vs. MA)** **(b) MazeWorld (vanilla vs. AM)**

**(c) Recycle (vanilla vs. MA)** **(d) Recycle (vanilla vs. AM)**

**Figure 5: Vanilla UCT vs. UCT with move splitting**

are randomly chosen as possible goals. Each potential goal location has a color blue/green and is of the shape circle/square. Each potential message from the set $\{blue, green, square, circle\}$ has an 80% availability probability. The cost for sending a message is randomly sampled from $[0.0, 0.25]$. The parameter $\lambda$ in the reward function is sampled from $(0.0, 1.0)$. For 30 of the problem instances, we used the negative TV distance from the target belief as the belief-dependent reward. For the remaining instances, we used the entropy of the observer's belief as the belief-dependent reward. This approach rewards obscuring the intended goal.

For the Recycle problem, we generated 30 random instances, each with five items initially placed in bins at random. Each potential message from the set $\{compost, recycle, trash\}$ has an 80% availability probability. Placing an object into a bin success with probability randomly sampled from $[0.3, 0.8]$. The cost for sending a message is randomly sampled from $[0.0, 0.5]$. The parameter $\lambda$ in the reward function is sampled from $(0.0, 1.0)$. The negative TV distance from the true belief was used as a belief-dependent reward.

**Effects of Move Splitting** Fig. 5 compares UCT (MA/AM) with the vanilla UCT. The result suggests that move splitting can improve the performance for some problem instances. This distinction is most evident within the Recycle scenario, as depicted in Fig. 5c-5d. While the effects of move splitting were less prominent in other configurations, move splitting did not hurt the performance except for a few problem instances.

## 8 RELATED WORK

Optimizing explicit communication behaviors has been previously explored in literature. For instance, Pynadath and Tambe [28] investigated a special case of MTDP with explicit communication. Similarly, Goldman and Zilberstein [15] examined an extension of Dec-POMDP [4] that included explicit communication actions. Studies in this area primarily focused on computing policies for all agents involved in the interaction, a perspective distinct from the human-robot interaction setting considered in this paper. The Communicating POMDP-IR (Com-POMDP-IR) model, proposed by [29], shares similarities with ours but also deviates in some important ways. While Com-OAMDP emphasizes explicit communication about the ego agent's types, Com-POMDP-IR focuses on conveying certain environmental aspects to the observer. It incorporates the observer's mental state as an additional state factor, maintaining a belief in this factor in a similar manner to POMDP. Sreedharan et al. [31] proposed an approach to combine explicability [22] and explicit communication.

Com-OAMDP could be regarded as a particular case of Decision Process with non-Markovian Reward (NMRDP) [2, 32], where rewards are non-Markovian. Unlike Com-OAMDPs, existing works on NMRDP [2, 5, 24, 32] utilize temporal logic to describe rewards over histories. Com-OAMDP, on the other hand, employs the belief function to capture the non-Markovian nature of rewards.

The concept of move-splitting has been leveraged to solve games with multiple actions [10, 18, 21] and MDPs with factored actions [12]. Com-OAMDP is another example of a problem where actions can naturally be partitioned into distinct components.

An intriguing question pertains to the recursion level necessary for modeling the observer. In the examples used in this paper, we only required the basic level of nested reasoning: in these instances, the Com-OAMDP agent is at the strategy level 2, aiming to influence an observer at level 1, under the assumption that the observer models the observed agent as level 0 (unaware of the observer). However, we could consider a Com-OAMDP agent at a higher strategy level. For instance, a Com-OAMDP agent at the strategy level 4 could optimally influence an observer at level 3, who assumes that the observed agent is at level 2. The Rational Speech Act Model [11, 16] suggests that the listener (observer) employs recursive thinking to comprehend utterances. Quantifying the role of this recursive thinking remains an area for future investigation.

## 9 CONCLUSION

In this paper, we present a computational model called Communicative Observer-Aware Markov Decision Process (Com-OAMDP). This model is tailored for planning both implicit and explicit communication of intentions, goals, and desires. Com-OAMDP builds upon OAMDP and focuses on optimally influencing the observer's mental state via the agent's actions and messages. We show that Com-OAMDPs can be seen as a special case of the Communicative Interactive Partially Observable Markov Decision Process (CIPOMDP), primarily concerning scenarios with full observability and passive observers. We propose a solution technique for Com-OAMDP based on splitting domain and communication actions. Our empirical evaluation illustrate the efficacy of solving Com-OAMDP problems using MCTS. An important avenue for future exploration involves the empirical evaluation of the observer's models via user studies, as well as extending the solution method to handle higher level of recursive thinking.

## ACKNOWLEDGMENTS

This research was supported in part by the NSF grant number IIS-2205153 and by the Alliance Innovation Lab Silicon Valley.

## REFERENCES

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the Multiarmed Bandit Problem. *Machine Learning* 47, 2 (2002), 235–256.

[2] Fahiem Bacchus, Craig Boutilier, and Adam Grove. 1996. Rewarding behaviors. In *Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence*, Vol. 2. 1160–1167.

[3] Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. 2009. Action understanding as inverse planning. *Cognition* 113 (2009), 329–349.

[4] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research* 27, 4 (2002), 819–840.

[5] Ronen Brafman, Giuseppe De Giacomo, and Fabio Patrizi. 2018. LTLf/LDLf non-Markovian rewards. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 1771–1778.

[6] Debargha Dey, Andrii Matviienko, Melanie Berger, Bastian Pfleging, Marieke Martens, and Jacques Terken. 2021. Communicating the intention of an automated vehicle to pedestrians: The contributions of eHMI and vehicle behavior. *Information Technology* 63, 2 (2021), 123–141.

[7] Anca Dragan, Rachel Holladay, and Siddhartha Srinivasa. 2015. Deceptive robot motion: Synthesis, analysis and experiments. *Autonomous Robots* 39, 3 (2015), 331–345.

[8] Anca D. Dragan, Kenton C. T. Lee, and Siddhartha S. Srinivasa. 2013. Legibility and predictability of robot motion. In *Proceedings of Eighth ACM/IEEE International Conference on Human-Robot Interaction*. 301–308.

[9] Jaime F. Fisac, Chang Liu, Jessica B. Hamrick, Shankar Sastry, J. Karl Hedrick, Thomas L. Griffiths, and Anca D. Dragan. 2020. Generating plans that predict themselves. In *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics (Springer Proceedings in Advanced Robotics)*. 144–159.

[10] David Fotland. 2006. Building a world-champion Arimaa program. In *Computers and Games (Lecture Notes in Computer Science)*. 175–186.

[11] Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science* 336, 6084 (2012), 998–998.

[12] Florian Geißer, David Speck, and Thomas Keller. 2020. Trial-based heuristic tree search for MDPs with factored action spaces. *Proceedings of the International Symposium on Combinatorial Search* 11 (2020), 38–47.

[13] Piotr Gmytrasiewicz. 2020. How to do things with words: A Bayesian approach. *Journal of Artificial Intelligence Research* 68 (2020), 753–776.

[14] Piotr J. Gmytrasiewicz and Doshi Prashant. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24 (2005), 49–79.

[15] Claudia V. Goldman and Shlomo Zilberstein. 2003. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. 137–144.

[16] Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences* 20, 11 (2016), 818–829.

[17] John C. Harsanyi. 1968. Games with incomplete information played by "Bayesian" players, I-III. Part III. The basic probability distribution of the game. *Management Science* 14, 7 (1968), 486–502.

[18] Niels Justesen, Tobias Mahlmann, Sebastian Risi, and Julian Togelius. 2018. Playing multiaction adversarial games: Online evolutionary planning versus tree search. *IEEE Transactions on Games* 10, 3 (2018), 281–291.

[19] Leslie Pack Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101, 1-2 (1998), 99–134.

[20] Anirudh Kakarlapudi, Gayathri Anil, Adam Eck, Prashant Doshi, and Leen-Kiat Soh. 2022. Decision-theoretic planning with communication in open multiagent systems. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. 938–948.

[21] Jakub Kowalski, Maksymilian Mika, Wojciech Pawlik, Jakub Sutowicz, Marek Szykuła, and Mark H. M. Winands. 2022. Split moves for Monte-Carlo tree search. *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (2022), 10247–10255.

[22] Anagha Kulkarni, Yantian Zha, Tathagata Chakraborti, Satya Gautam Vadlamudi, Yu Zhang, and Subbarao Kambhampati. 2019. Explicable planning as minimizing distance from expected behavior. In *International Conference on Autonomous Agents and MultiAgent Systems*. 2075–2077.

[23] Minae Kwon, Sandy H. Huang, and Anca D. Dragan. 2018. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 87–95.

[24] Michael L. Littman, Ufuk Topcu, Jie Fu, Charles Isbell, Min Wen, and James MacGlashan. 2017. Environment-independent task specifications via GLTL. *arXiv:1704.04341* (2017).

[25] Peta Masters and Sebastian Sardina. 2017. Deceptive path planning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 4368–4375.

[26] Shuwa Miura, Andrew Cohen, and Shlomo Zilberstein. 2021. Maximizing legibility in stochastic environments. In *Proceedings of the Thirtieth IEEE International Conference on Robot and Human Interactive Communication*. 1053–1059.

[27] Shuwa Miura and Shlomo Zilberstein. 2021. A unifying framework for observer-aware planning and its complexity. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. 610–620.

[28] David V. Pynadath and Milind Tambe. 2002. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research* 16, 1 (2002), 389–423.

[29] Jennifer Renoux, Tiago Veiga, Pedro Lima, and Matthijs Spaan. 2020. A unified decision-theoretic model for information gathering and communication planning. In *Proceedings of the Twenty-Ninth IEEE International Conference on Robot and Human Interactive Communication*. 67–74.

[30] Sven Seuken and Shlomo Zilberstein. 2008. Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multi-Agent Systems* 17, 2 (2008), 190–250.

[31] Sarath Sreedharan, Tathagata Chakraborti, Christian Muise, and Subbarao Kambhampati. 2020. Expectation-aware planning: A unifying framework for synthesizing and executing self-explaining plans for human-aware planning. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 03 (2020), 2518–2526.

[32] Sylvie Thiébaux, Charles Gretton, John Slaney, David Price, and F. Kabanza. 2006. Decision-theoretic planning with non-Markovian rewards. *Journal of Artificial Intelligence Research* 25 (2006), 17–74.