Robust Active Measuring under Model Uncertainty

Merlijn Krale[1], Thiago D. Simão[2], Jana Tumova[3], Nils Jansen[1,4]
[1] Radboud University Nijmegen, the Netherlands,
[2] Eindhoven University of Technology, the Netherlands,
[3] KTH Royal Institute of Technology, Stockholm, Sweden,
[4] Ruhr-University Bochum, Germany
merlijn.krale@ru.nl, t.simao@tue.nl, tumova@kth.se, nils.jansen@ru.nl

Abstract

Partial observability and uncertainty are common problems in sequential decision-making that particularly impede the use of formal models such as Markov decision processes (MDPs). However, in practice, agents may be able to employ costly sensors to *measure* their environment and resolve partial observability by gathering information. Moreover, imprecise transition functions can capture model uncertainty. We combine these concepts and extend MDPs to *robust active-measuring MDPs (RAM-MDPs)*. We present an active-measure heuristic to solve RAM-MDPs efficiently and show that model uncertainty can, counterintuitively, let agents take fewer measurements. We propose a method to counteract this behavior while only incurring a bounded additional cost. We empirically compare our methods to several baselines and show their superior scalability and performance.

Introduction

Markov decision processes (MDPs; Puterman 1994) are a standard sequential decision-making model (Kormushev, Calinon, and Caldwell 2013; Lei et al. 2020; Sunberg and Kochenderfer 2022). However, in MDPs, the decision-maker has full knowledge of the dynamics of the environment and its current state, which is often unrealistic. Well-studied frameworks exist to relax these assumptions. To represent model uncertainty, *robust MDPs* (RMDPs; Nilim and Ghaoui 2005) extend MDPs by replacing transition probabilities with uncertainty sets. To represent state uncertainty, *partially observable MDPs* (POMDPs; Kaelbling, Littman, and Cassandra 1998) extend MDPs with an observation function, which dictates how the agent gains information while interacting with the environment.

Active-measure MDPs are a subset of the latter, where agents have direct control over when and how they gather information, which has an associated cost (Bellinger et al. 2021). For example, a drone may request information from a motion capture system which has costs related to communication (Figure 1). Furthermore, this model can capture applications in predictive maintenance and healthcare, such as diagnostics (Jimenez-Roa et al. 2022; Yu et al. 2023). In these applications, the cost or risk of gaining information needs to be weighed against the value of obtaining more information.

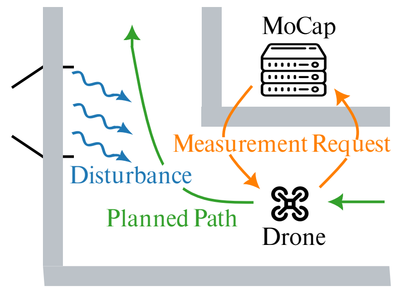

Figure 1: A motivating example. A drone has to plan a path through a corridor where (wind) disturbances introduce uncertainty in its position. An external Motion Capture (MoCap) system can provide the drone's exact position, but this uses some of its limited bandwidth. How should the risk of a collision be weighed against the cost of using this system?

Settings with both model and state uncertainty can be expressed as *robust POMDPs* (RPOMDPs; Osogami 2015). However, even though uncertain and partially observable settings have been studied extensively on their own, research on RPOMDPs has been limited in part due to their complexity. Existing strategies for solving RPOMDPs are either exact but computationally expensive (Osogami 2015; Rasouli and Saghafian 2018), or only consider policies with limited memory (Suilen et al. 2020; Cubuktepe et al. 2021).

Aiming to achieve better performance and scalability, this paper focuses on a subset of RPOMDPs with *active measuring*, which we formally define as *robust active-measuring MDPs* (RAM-MDPs). We make the counter-intuitive observation that high model uncertainty may discourage measuring in certain environments. For solving a specific subset of RAM-MDPs, we adopt a heuristic called *act-then-measure* (ATM; Krale, Simão, and Jansen 2023) for standard active-measure environments in an uncertain setting. This heuristic suggests partially ignoring future state uncertainty, which drastically decreases policy computation times.

Next, we propose *measurement leniency*, a strategy to encourage measuring in settings with high model uncertainty. This strategy allows the agent to make additional measurements when this would yield better results under a less pessimistic model. We formalize this idea and prove that measurement-lenient policies have a bounded lost return as

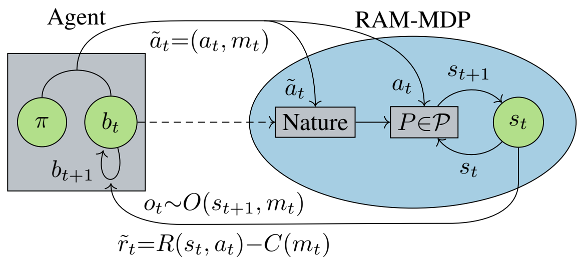

Figure 2: A visualization of agent-environment interactions in RAM-MDPs.

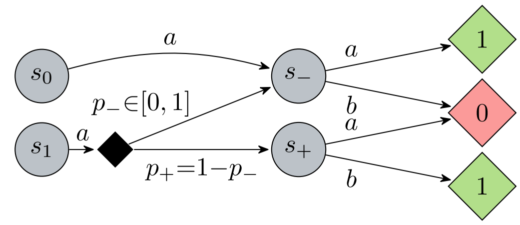

Figure 3: For initial belief b over s_0 and s_1 in this RPOMDP, the expected return is minimized if, for the next belief b', probabilities of being in state s_- and s_+ are equal. The corresponding values of p_- and p_+ depend on $b(s_0)$ and $b(s_1)$, thus, the worst-case transition function is belief-dependent.

compared to their fully robust counterpart.

We empirically compare both regular and measurement-lenient variants of our algorithm on a number of environments. Against a number of baselines, we demonstrate (1) the computational tractability of our method and (2) an increased robustness of policies.

Contributions. The main contributions of this work are: (1) defining RAM-MDPs to represent active measuring in uncertain environments; (2) analyzing the influence of model uncertainty on measuring behavior; (3) showing how the *act-then-measure* heuristic can be used to efficiently solve a subset of RAM-MDPs; and (4) defining the *measurement leniency* strategy to get better performance in settings with high model uncertainty.

Setting: RAM-MDPs

In this section, we formally define RAM-MDPs as the combination of the RMDP (Nilim and Ghaoui 2005) and active-measure (Bellinger et al. 2021) frameworks:

Definition 1. A *robust active-measure MDP* (RAM-MDP) is a tuple $\mathcal{M} = (S, R, \gamma, \tilde{A} = A \times M, \mathcal{P}, \mathcal{I}, O, \Omega, C)$, with state space S, reward function $R \colon S \times A \to \mathbb{R}$, and discount factor γ. $\tilde{A}$ is the set of *actions*, which consists of pairs of *control* and *measurement* actions $\tilde{a} = \langle a, m \rangle \in A \times M$. Control actions affect the environment, while measurement actions affect what information agents gain about their current state. The dynamics are given by an *uncertain transition function* $\mathcal{P} \colon S \times A \times S \to \mathcal{I}$, which gives an interval from the *interval set* $[p_{\min}, p_{\max}] \in \mathcal{I}$ (with $0 \leq p_{\min} \leq p_{\max} \leq 1$) for each transition. Like POMDPs, RAM-MDPs have an *observation function* $O \colon S \times M \times \Omega \to \mathbb{R}$ with Ω the observation space. Lastly, the cost of measuring is given by $C \colon M \to \mathbb{R}$.

RAM-MDPs are a subset of RPOMDPs, with the additional property that the action space is factorized into control- and measuring actions, and O and $\mathcal{P}$ are independent of these respective action types. Furthermore, RAM-MDPs collapse to RMDPs if all measurements have cost 0 and a unique observation for all states, and to MDPs if, in addition, $p_{\min} = p_{\max}$ for all intervals in $\mathcal{I}$, and $\mathcal{P}$ forms a valid probability distribution for all state-action pairs.

Inspired by Nam, Fleming, and Brunskill (2021), we assume that measurements are *complete* and *noiseless*. Intuitively, this means agents only have two measurement options: they either take a measurement that returns full state information, or they take no measurement. In this case, the observation set has the form $\Omega \colon S \cup \{\perp\}$, and the observation function is deterministic, with $O \colon S \times M \to \Omega$, such that $\forall s \colon O(\perp | s, 0) = 1$ and $O(s|s, 1) = 1$. Lastly, we assume $C(0) = 0$ and denote $C(1) = c$.

Agent-environment interactions for RAM-MDPs can be viewed as a two-player game between the agent and 'nature,' as visualized in Figure 2. Starting from an initial state s_0, for each timestep t the agent chooses an action pair $\tilde{a}_t = \langle a_t, m_t \rangle$ to execute according to some policy π. Based on the chosen action pair, the current state, and the agent's current belief, nature picks a valid probability function $P(\cdot | s_t, \tilde{a}_t)$ from the uncertainty set, i.e., subject to the constraint that $\forall s' \colon P(s' | s_t, \tilde{a}_t) \in \mathcal{P}(s_t, a_t, s')$ and $\sum_{s' \in S} P(s' | s_t, \tilde{a}_t) = 1$. Then, the environment transitions to a new state $s_{t+1} \sim P(\cdot | s_t, a_t)$, and returns a scalarized reward $\tilde{r}_t = R(s_t, a_t) - C(m_t)$ and observation $o_t \sim O(\cdot | s_{t+1}, m_t)$ to the agent. The goal of the agent is to compute a policy π with the highest expected discounted scalarized return. We assume these policies are belief-based, that is, $\pi \colon b \to \tilde{A}$ for a belief $b \in \Delta(S)$ over states.

To make this problem more tractable, we make a few assumptions. Our description of the agent-environment interactions already assumes full observability for nature, as well as a dynamic (Nilim and Ghaoui 2005) and (s, a)-rectangular (Wiesemann, Kuhn, and Rustem 2013) system. These assumptions mean that the transition probabilities picked by nature may be different at each timestep and are independent of other transition probabilities, which are both common assumptions in RPOMDP literature. Next, we assume nature is *adversarial*, meaning it chooses transition functions to minimize the expected discounted scalarized return of the agent. As for RPOMDPs, these assumptions mean worst-case transition probabilities are generally *belief-dependent*, as shown by the RPOMDP in Figure 3. Thus, the worst-case transition- and value functions, P_{R} and V_{R}, are both belief-dependent. We first introduce $b'_{\text{R}}(b, \tilde{a})$ as the expected distribution over states in the next step when taking action pair $\tilde{a} \in \tilde{A}$ in belief b:

$$b'_{\text{R}}(s' | b, \tilde{a}) = \sum_{s \in S} b(s) P_{\text{R}}(s' | s, b, \tilde{a}). \tag{1}$$

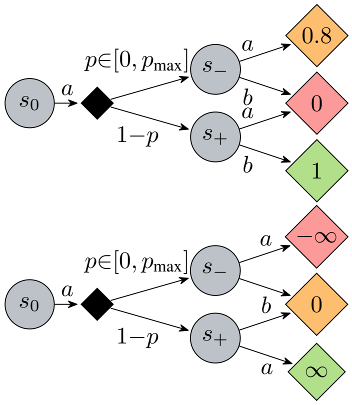

Figure 4: Environments A-B (top) and LUCKY-UNLUCKY (bottom).

Using this notation, we define P_R as follows:

$$P_R(s'|s,b,\langle a,m\rangle) = \underset{P_R\in\mathcal{P}(s,a,\cdot)}{\arg\inf}\; V_R\big(b'_R(b,\langle a,m\rangle)\big), \qquad (2)$$

where we note that the minimization affects both V_R directly, as well as b'_R. With this, V_R is given as follows:

$$V_R(b) = \max_{\tilde{a}=\langle a,m\rangle\in\tilde{A}} \sum_{s\in S} b(s)\Big(R(s,a) - C(m) + \gamma\sum_{s'\in s} P_R\big(s'|s,b,\tilde{a}\big)V_R\big(b'_R(b,\tilde{a})\big)\Big). \qquad (3)$$

RAM-MDP Properties

In this section, we highlight and discuss a number of interesting properties of RAM-MDPs.

The worst-case transition function is measurement-dependent. Equation (2) defines a worst-case transition function that depends on the complete action pair $\tilde{a}=\langle a,m\rangle$ rather than only on the control action a. Thus, even though the uncertain transition function is independent of what measurement is chosen, the worst-case transition function is not.

As an example of why this dependency holds, consider the A-B RAM-MDP (Figure 4 top). This environment has three states: an initial state s_0, and two next states s_- and s_+ with different optimal actions a and b. However, the reward for taking the optimal action in s_- is lower than that in s_+. We are interested in finding the worst p when we measure in s_0 and when we do not. When measuring, s_- has a lower expected value than s_+, meaning the worst-case transition has $p=1$. When not measuring, however, this deterministic transition means the agent can safely pick action a and receive a reward of 0.8. Instead, if p is chosen closer to 0.5, the expected return of taking action a decreases, which gives worse expected returns overall. Thus, the worst-case transition function depends on the chosen measuring action.

Intuitively, we find that for fully observable transitions (such as when measuring), the worst-case is simply given by maximizing worst-case outcomes. However, for partially observable transitions (such as when not measuring), an *unpredictable* outcome is often worse since this requires considering all possible outcomes for the next action.

Algorithm 1: ROBUST ATM PLANNER

Pre-compute P_{RMDP} and Q_{RMDP}
Initialise $b_0(s) = \delta(s, s_0)$
while episode not completed **do**
  Pick control action a_t ▷ *Equation* (4)
  Pick corresponding measuring action m_t ▷ *Equation* (6)
  Execute $\tilde{a}_t = \langle a_t, m_t\rangle$
  Receive reward $\tilde{r}_t$ and observation o_t
  Determine next worst-case belief b_{t+1} ▷ *Equation* (1)
return $\sum_t \gamma^t \tilde{r}_t$

High uncertainty can discourage measuring. The assumption that nature is adversarial (and thus chooses worst-case outcomes) is common for RMDPs. However, in partially observable settings, nature does not only influence future predictions (via P_R) but (importantly) also predictions of past interactions (via b_R), and thus the current belief. This may lead to overly conservative beliefs, especially if uncertainty is high.

As an example of overly conservative behavior induced by such beliefs, consider the LUCKY-UNLUCKY RAM-MDP (Figure 4 bottom). As before, this environment has three states s_0, s_+, and s_-, where we interpret the latter two as a lucky and unlucky state. In both, taking 'safe' action b leads to a neutral reward, while taking 'risky' action a gives an infinite positive reward in s_+ and an infinite negative reward in s_-. We are interested in measuring behavior at different uncertainty intervals, as specified by $p_{\max}$. We notice the expected returns for s_- are strictly lower than those of s_+, meaning an adversarial nature always chooses the highest possible probability p, regardless of whether the agent chooses to measure. First, we assume $p_{\max}=1$. This means the transition is deterministic, in which case measuring gives no additional information but still incurs a measuring cost and is thus sub-optimal. Next, we assume $p_{\max}<1$. In this case, it is optimal to measure since this means spending a (finite) measuring cost to possibly achieve an infinite reward. Counterintuitively, we find that high model uncertainty may lead to optimal policies taking fewer measurements than if model uncertainty is lower, even if measuring would alleviate this uncertainty. This property occurs even for finite returns and non-zero probabilities, as shown in Appendix B (Krale et al. 2023).

The described behavior is the result of optimal robust policies (over-) optimizing for the worst case while not considering other possible outcomes. This behavior makes sense in contexts where the environment must be considered adversarial, such as in security settings. However, if policies are required to perform well on all possible models, such as when uncertainty represents confidence intervals, this over-optimization is unwanted behavior. Moreover, if observations have additional value not captured by the model, such as to improve the model itself, we would want our policies to take measurements more leniently. We will introduce a method to encourage such leniency later in this paper.

Act-Then-Measure in RAM-MDPs

In this section, we describe a method for finding approximate solutions for RAM-MDPs in a computationally tractable manner. In particular, we extend the ATM heuristic (Krale, Simão, and Jansen 2023) to an uncertain setting.

> **The robust act-then-measure (RATM) heuristic:** (1) chooses control-actions assuming that all (state) uncertainty will be resolved *in the next* states (after one time-step); and (2) chooses measuring-actions and updates beliefs assuming that all (state) uncertainty will be resolved *after the next* states (after two time-steps).

Since measurement actions only affect future state uncertainty, the first point of the heuristic allows us to *pick control actions independently from measuring actions*. Thus, our high-level strategy is given by Algorithm 1. The remainder of this section will explain in detail how to perform each step in this algorithm.

Choosing Control Actions

Similar to the Q_{MDP} heuristic (Littman, Cassandra, and Kaelbling 1995), the RATM heuristic means that returns of control actions can be approximated by those of the underlying RMDP:

$$Q_{\text{RATM}}(b,a) = \sum_{s\in S} b(s) Q_{\text{RMDP}}(s,a) \approx Q_{\text{R}}(b,a), \tag{4}$$

where Q_{RATM} denotes the approximate expected value when following the RATM heuristic, and Q_{RMDP} and Q_{R} denote optimal expected values for the RAM-MDP and its underlying RMDP, respectively. Generally, Q_{RMDP} can be efficiently pre-computed, allowing for faster policy computations than methods that fully consider partial observability.

Computing Measuring Value

Next, we need a method to pick measurement actions. With noiseless and complete measurements, we define the *robust measuring value* MV_{RATM} as the difference in expected value between measuring and non-measuring actions:

$$\text{MV}_{\text{RATM}}(b,a) = Q_{\text{RATM}}(b,\langle a,1\rangle) - Q_{\text{RATM}}(b,\langle a,0\rangle). \tag{5}$$

Measuring is optimal for positive measuring values only, which yields the following measuring condition:

$$m_{\text{RATM}}(b,a) = \begin{cases} 1 & \text{if } \text{MV}_{\text{RATM}}(b,a) \geq 0; \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

To compute MV_{RATM}, we need expressions for $Q_{\text{RATM}}(b,\langle a,1\rangle)$ and $Q_{\text{RATM}}(b,\langle a,0\rangle)$. We note that the RATM heuristic means that for both, Equation (4) can be applied to all beliefs *after* the next one. Thus, the Q-value when measuring is given as:

$$\begin{aligned} Q_{\text{RATM}}(b,\langle a,1\rangle) &= R(b,a) - c + \gamma \sum_{s} b(s) \\ &\Big[\sum_{s'} P_{\text{R}}(s'|s,\langle a,1\rangle,b) \max_{a} Q_{\text{RMDP}}(s',a)\Big], \end{aligned} \tag{7}$$

with $R(b,a) = \sum_s b(s)R(s,a)$. Here, we decide the next actions for each state separately, which we can only do if we take a measurement. When not measuring, we must instead pick an optimal action considering all possible next states:

$$\begin{aligned} Q_{\text{RATM}}(b,\langle a,0\rangle) &= R(b,a) + \gamma \sum_{s} b(s) \\ &\Big[\max_{a} \sum_{s'} P_{\text{R}}(s'|s,\langle a,0\rangle,b) Q_{\text{RMDP}}(s',a)\Big]. \end{aligned} \tag{8}$$

Combining both equations, we write the robust measuring value of Equation (5) as follows:

$$\begin{aligned} \text{MV}_{\text{RATM}}(b,a) &= -c + \max_{a'\in A} \gamma \sum_{s\in S} b(s) \\ &\Big[Q_{\text{RMDP}}(s,a) - \sum_{s'\in S} P_{\text{R}}(s'|s,\langle a,0\rangle,b) Q_{\text{RMDP}}(s',a')\Big]. \end{aligned} \tag{9}$$

We notice that this equation contains only belief-independent and thus pre-computable quantities, with the exception of the transition function P_{R}. This function is equal to P_{RMDP} when measuring and otherwise given as:

$$\begin{aligned} P_{\text{R}}(s'|s,\langle a,0\rangle,b) &= \max_{a_b\in A} \min_{P(\cdot|s,a)\in\mathcal{P}(s,a,\cdot)} \sum_{s} b(s) \\ &\Big[\sum_{s'\in S} P(s'|s,a) Q_{\text{RMDP}}(s',a_b)\Big]. \end{aligned} \tag{10}$$

This equation can be tractably solved by a (non-convex) mixed integer program (MIP). However, since this problem needs to be solved at every step, we find these computations take up the majority of the (online) runtime in our experiments. With all quantities defined, we have fully described all steps in Algorithm 1. Alternatively, we may define this algorithm as a policy:

$$\pi_{\text{RATM}}(b) = \langle \max_{a\in A} Q_{\text{R}}\big(b,a\big), m_{\text{RATM}}\big(b, \max_{a\in A} Q_{\text{R}}(b,a)\big)\rangle. \tag{11}$$

Measurement Leniency

As outlined previously, robust policies can exhibit (overly) conservative measuring behavior in RAM-MDPs, particularly if model uncertainty is high. In this section, we propose *measurement leniency* to counteract this behavior. Intuitively, measurement leniency means that agents choose control actions to optimize for the worst case, but may take extra measurements according to less pessimistic metrics, such as average expected returns. Since the cost of extra measurements is bounded and predictable, measurement leniency might give sufficient robustness guarantees for many real-life applications while allowing less conservative behavior. We formally define measurement leniency as follows:

Definition 2. Let π be any policy. A corresponding *measurement-lenient* policy is any policy π_{ML} such that:

(1) $\forall b, \pi(b) = \langle a,1\rangle \implies \pi_{\text{ML}}(b) = \langle a,1\rangle$, and
(2) $\forall b, \pi(b) = \langle a,0\rangle \implies \pi_{\text{ML}}(b) = \langle a,m\rangle$.

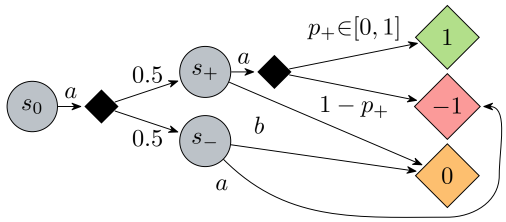

Figure 5: A RAM-MDP where the measuring value of $\mathcal{M}_{\text{ML}}$ for a measurement-lenient policy is sub-optimal. Assuming an adversarial nature, the optimal control action in the states s_+ and s_- is b, meaning measuring in s_0 is sub-optimal. However, choosing a $\mathcal{M}_{\text{ML}}$ with $p_+ > 0.5$ would yield a positive measuring value.

Using this definition, we define an optimal measurement-lenient policy as one that maximizes the expected discounted scalarized return in some (less conservative) environment $\mathcal{M}_{\text{ML}}$. For example, if a probability distribution over transition functions is known, $\mathcal{M}_{\text{ML}}$ could represent the most likely outcomes. We formalize this as follows:

Definition 3. Let π be a policy for an RAM-MDP $\mathcal{M}$, and Π_{ML} the corresponding set of measurement-lenient policies. Further, let $\mathcal{M}_{\text{ML}}$ be any active-measure model with the same state- and action space as $\mathcal{M}$. The *optimal measurement-lenient* policy π^*_{ML} with respect to $\mathcal{M}_{\text{ML}}$ is given as:

$$\pi^*_{\text{ML}} = \arg\max_{\pi_{\text{ML}} \in \Pi_{\text{ML}}} \mathbb{E}_{\pi_{\text{ML}}, \mathcal{M}_{\text{ML}}} \sum_t \gamma^t \tilde{r}_t \tag{12}$$

Computing Measurement-lenient Policies

By construction, control actions of measurement-lenient policies are equal to those in their base policies. For any belief b, we may thus assume a control action $a_{\text{R}}(b)$ is given.

In order to make optimal measuring choices, we need to keep track of the current belief according to the dynamics of both $\mathcal{M}$ and $\mathcal{M}_{\text{ML}}$. For the latter, we denote b_{ML} as the current belief and $b'_{\text{ML}}(b_{\text{ML}}, \tilde{a})$ as the belief after taking action $\tilde{a}$ in belief b_{ML}. However, as shown in Figure 5, simply using this belief to compute generic measuring value for $\mathcal{M}_{\text{ML}}$ does not yield the correct behavior since this does not take into account that future control actions will be based on $\mathcal{M}$ instead. To account for this, we first define the Q-value function for following a measurement-lenient policy in $\mathcal{M}_{\text{ML}}$:

$$\begin{aligned} &Q_{\text{ML}}(b_{\text{ML}}, b, \langle a, m\rangle) = R(b_{\text{ML}}, a) - C(m) \\ &+\gamma \max_{m' \in M} Q_{\text{ML}}\Big(b'_{\text{ML}}(b_{\text{ML}}, \tilde{a}), b'_{\text{R}}(b, \tilde{a}), \langle a_{\text{R}}(b'_{\text{R}}(b, a)), m'\rangle\Big) \end{aligned} \tag{13}$$

For complete and noiseless measurements, we express the measuring value for measurement-lenient policies as:

$$\text{MV}_{\text{ML}}(b_{\text{ML}}, b, a) = Q_{\text{ML}}(b_{\text{ML}}, b, \langle a, 1\rangle) - Q_{\text{ML}}(b_{\text{ML}}, b, \langle a, 0\rangle) \tag{14}$$

We first note that after a measurement b'_{R} and b'_{ML} are always equal to the observation that has been made. Thus, the Q-value when measuring can be expressed as follows:

$$\begin{aligned} &Q_{\text{ML}}(b_{\text{ML}}, b, \langle a, 1\rangle) = R(b_{\text{ML}}, a) - c \\ &\quad + \gamma \sum_{s \in S} b'_{\text{ML}}(s | b_{\text{ML}}, \tilde{a}) \max_{m' \in M} Q_{\text{ML}}(s, s, \langle a_{\text{R}}(s), m'\rangle) \end{aligned} \tag{15}$$

Unfortunately, there is no trivial way to simplify our expression for non-measuring actions without making further assumptions about the policy choosing control actions. For the general case where the fully robust policy is unknown, we thus propose to approximate our Q-values using the robust act-then-measure heuristic, as defined in the previous section. We restate the relevant part as follows:

> **Measurement-lenient approximation**: Choose measurement-lenient measuring actions assuming all (state) uncertainty will be resolved *after* the next state.

Using this approximation, we can simplify our equations by replacing future Q-values with those of the fully observable variant of the environment. We denote this Q-value as Q_{CRMDP} and rewrite Equation (13) as follows:

$$\begin{aligned} &Q_{\text{ML}}(b_{\text{ML}}, b, \langle a, m\rangle) \approx R(b_{\text{ML}}, a) - C(m) \\ &\quad + \gamma \sum_{s' \in S} Q_{\text{CRMDP}}\Big(s', a_{\text{R}}\big(b'_{\text{R}}(b, \langle a, m\rangle)\big)\Big) \end{aligned} \tag{16}$$

We note that this expression does not require solving an optimization problem, meaning it can be computed quickly. With this, we approximate Equation (14) as follows:

$$\begin{aligned} &\text{MV}_{\text{ML}}(b_C, b, a) \approx -C(m) + \gamma \max_{a_b \in A} \sum_{s \in S} b_{\text{ML}}(s) \\ &\quad \Big[\max_{a \in A} Q_{\text{CRMDP}}(s', a) - Q_{\text{CRMDP}}(s', a_{\text{R}}(b'_{\text{R}}(b, a)))\Big] \end{aligned} \tag{17}$$

For measurement-lenient policies, the measuring condition requires the measuring value of both $\mathcal{M}_{\text{ML}}$ and $\mathcal{M}$ to be non-negative, which gives:

$$m_{\text{ML}}(b_{\text{ML}}, b, a) = \begin{cases} 1 & \text{if } \text{MV}_{\text{ML}}(b_{\text{ML}}, a) \geq 0 \\ & \text{or } \text{MV}_{\text{R}}(b, a) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Regret of Measurement-lenient Policies

One obvious downside of using measurement-lenient policies is that their worst-case performance is generally lower. However, we can show that their performance loss, as compared to their base policy, is bounded. Intuitively, this bound follows from the fact that measurement-lenient policies only take extra measurements, which decrease the total expected returns by (at most) c per step. We state this more formally:

Theorem 1. *Given an RAM-MDP $\mathcal{M}$ with complete and noiseless measurements and policy π. For any corresponding measurement-lenient policy π_{ML}, the following holds*

$$\forall b : V(\pi, b) - V(\pi_{ML}, b) \leq \sum_{n=0}^{\infty} \gamma^n c \tag{19}$$

We prove this theorem in Appendix B (Krale et al. 2023).

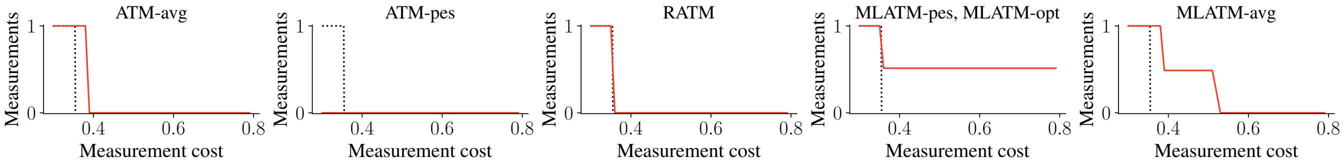

Figure 6: Mean number of measurements in A-B environment against measuring cost. Dotted lines show optimal behavior.

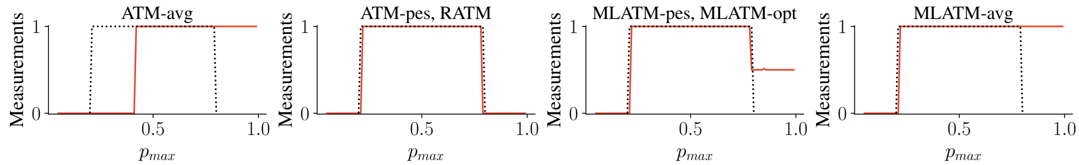

Figure 7: Mean number of measurements in LUCKY-UNLUCKY environment against $p_{\max}$. Dotted lines show optimal behavior.

Empirical Analysis

This section presents an empirical analysis of the behavior and performance of the proposed methods. We run experiments on (1) the A-B and LUCKY-UNLUCKY environments (Figure 4); and (2) two larger custom environments called SNAKEMAZE and DRONE that we describe in more detail below. We test the following algorithms:

- RATM: the robust planning algorithm following the RATM heuristic, as described in Algorithm 1.
- MLATM: the measurement-lenient variant of RATM, considering three choices of $\mathcal{M}_{\text{ML}}$: an *optimistic*, a *pessimistic*, and an *average*, denoted MLATM-*opt*, MLATM-*pes*, and MLATM-*avg*, respectively. We provide full definitions in Appendix A (Krale et al. 2023).
- ATM: as a baseline, we use the ATM planner (Krale, Simão, and Jansen 2023). We test two variants, denoted by ATM-*avg* and ATM-*pes*, which plan on the average-case environment and on the environment with transition function P_{RMDP}.

We provide code at doi.org/10.5281/zenodo.10406844.

Behavior Evaluation

We start with two small-scale experiments to determine (1) if our algorithms consider the effect of measuring on the transition function; and (2) how these algorithms change measuring behavior for different sizes of uncertainty intervals. For this, we run all algorithms on both the A-B and LUCKY-UNLUCKY environments (Figure 4) and compare the algorithms with optimal behavior (depicted by dotted lines in the results), as calculated in Appendix A (Krale et al. 2023).

Figure 6 shows that ATM-*avg* and ATM-*pes* do not measure optimally in the A-B environment, which shows that they do not incorporate the effect of measuring on the transition function. In contrast, RATM measures optimally in this environment, while all measurement-lenient variants take more measurements than optimal[1]. Figure 7 shows that all algorithms except ATM-*avg* measure optimally when uncertainty is low, while the measurement-lenient variants make (sub-optimal) measurements for high uncertainty.

Figure 8: A 2×5 visualization of the SNAKEMAZE environment. An agent traverses a snaking maze from the blue initial stat to the green goal state. It moves via *safe* (grey) and *risky* (yellow) actions with different stochastic effects.

Performance Evaluation

Next, we test the performance of our algorithms on a custom environment called SNAKEMAZE, which is designed to require conservative behavior. As the visualization in Figure 8 shows, the agent has to traverse a snaking maze, receiving a reward of 1 for reaching the goal state and a small penalty for each prior step. To do so, the agent can choose a *safe* or *risky* action for each cardinal direction. Safe actions have a 0.5 chance to move the agent one or two steps in the given direction, while risky actions have a 0.6 chance to move three steps but a 0.4 chance to not move at all. Thus, risky actions move the agent further on average, but even a little uncertainty in the transition probabilities changes this.

Following Osogami (2012), we parameterize uncertainty with a *confidence level* $\alpha \in (0, 1]$. We define our uncertainty such that any transition probability is at most a factor $1/\alpha$ larger than for some base transition function P, i.e., $\forall s, s' \in S, a \in A\colon \mathcal{P}(s, a, s') = [0, 1/\alpha P(s'|s, a)]$. Thus, $\alpha = 1$ represents no model uncertainty, while uncertainty increases as α approaches 0. Since finding the exact worst-case transition function for a RAM-MDP is intractable, we instead test our algorithms on the *worst-case transition function assuming full observability*, i.e., using the transition function of the underlying RMDP. This means that measurement-dependent worst-case transitions (as in the A-B environment) never occur, and we expect ATM-*pes* (which optimizes for the RMDP environment) to outperform the other algorithms.

Figure 9 (left) shows the scalarized returns of the algorithms in the 10×10 SNAKEMAZE environment at different

[1] Surpisingly, some algorithms show non-deterministic measuring behavior. We explain this in Appendix A (Krale et al. 2023).

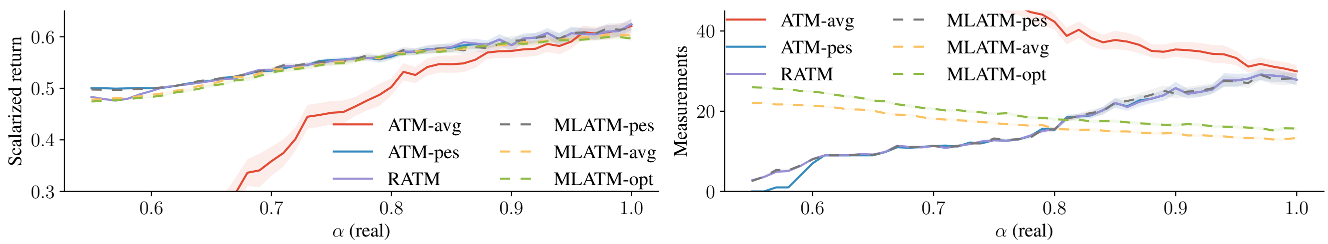

Figure 9: Returns and number of measurements in the SNAKEMAZE environment, with c=0.01. Uncertainty is parameterized by confidence level α and decreases left-to-right. Lines show the mean of 50 runs, with 95% confidence intervals shaded.

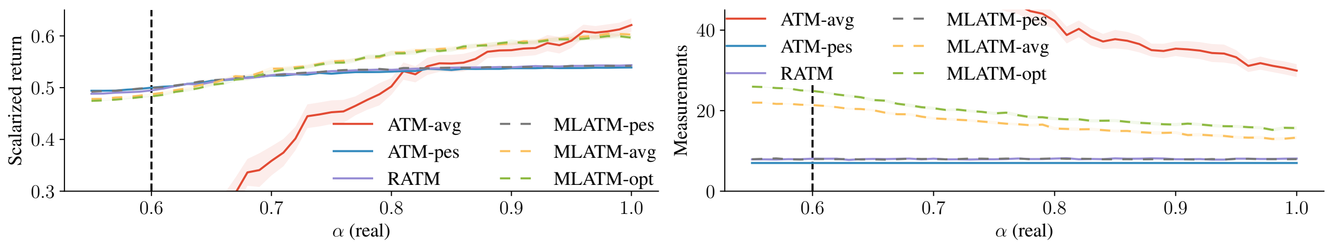

Figure 10: Returns and number of measurements in the SNAKEMAZE environment, with $c = 0.01$. Algorithms plan at uncertainty parametrized by $\alpha_p = 0.6$, but real dynamics is parametrized by α. Thus, real uncertainty decreases left-to-right, while planning uncertainty is given by the dotted line. Lines show the mean of 50 runs, with 95% confidence intervals shaded.

confidence levels α. We see that ATM-*avg* is outperformed by all other algorithms, while for α<0.7 MLATM-*avg* and MLATM-*opt* perform slightly worse than the more conservative algorithms. This difference is caused by their different measuring behavior, as shown on the right.

Next, we are interested in how the algorithms perform if uncertainty is misspecified, i.e., if the algorithms plan with a different uncertainty than that of the real environment. To test this, we let the algorithms plan on the SNAKEMAZE environment with uncertainty parameterized by α_p=0.6, while we deploy the algorithms on the environment with $\alpha \in [0.55, 1]$. Figure 10 (left) shows the scalarized returns of the different algorithms. We now see the advantage of measurement leniency: although MLATM-*avg* and MLATM-*opt* still perform slightly worse for large uncertainty, they outperform the more conservative algorithms for α>0.7. This can be explained by the algorithms taking more measurements (as shown on the right), which allows them to take advantage of the more favorable environment.

Scalability Evaluation

Lastly, to show that our algorithms scale to larger environments, we run all algorithms on a custom DRONE environment inspired by the example of Figure 1. A full explanation of this environment is given in Appendix A (Krale et al. 2023). The environment is a simplified and discretized motion model on a 2D grid, with $|S|$=39,204 states, $|A|$=25 actions, and up to 25 successor states per state-action pair. The goal of the agent is to reach a set of goal states without hitting any walls. Like before, we parameterize uncertainty using confidence levels and approximate the worst-case transition function by that of the underlying RMDP.

Figure 11 shows both the scalarized (left) and non-scalarized (right) returns of the algorithms at different confidence levels α. Similar to the results on SNAKEMAZE, ATM-*avg* performs worse than the other algorithms, while ATM-*pes* slightly outperforms all (control) robust algorithms in terms of scalarized returns. However, our (partially) robust algorithms outperform both ATM-*avg* and ATM-*pes* in terms of non-scalarized returns (i.e. returns ignoring measuring costs). Thus, our algorithms reach the goal states more often but take more measurements to do so. For misspecified uncertainty, as shown in Figure 12, we find similar results. All algorithms except ATM-*avg* obtain similar scalarized returns, while our algorithms outperform the baselines in terms of non-scalarized returns.

Interestingly, we do not find a difference between robust and measurement-lenient algorithms in this environment. To understand why, we first consider the SNAKEMAZE environment. Here, we notice that (1) an optimal agent only measures to know whether or not it can move down a row; and (2) worst-case outcomes are those that move the agent less far. Combined, this means higher uncertainty leads to *less frequent* measuring. In contrast, in the DRONE environment, the worst-case outcomes are generally those that bring the agent closer to a wall. Thus, for high uncertainty, agents already take *more frequent* measurements to prevent collisions, meaning measurement leniency has less impact.

Discussion

Finally, we provide a summary of our findings.

R(C)ATM considers the effect of measuring. RATM performs optimally in the A-B environment, which is only possible when considering the effect of measuring. The control-robust variants only perform sub-optimally by taking more measurements, as expected.

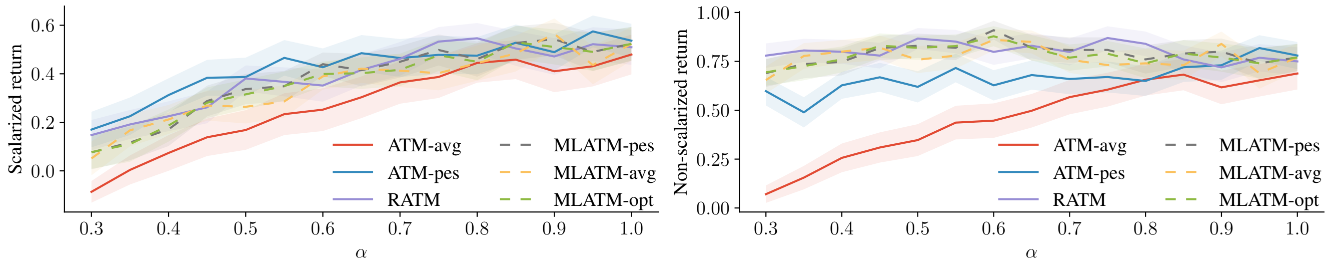

Figure 11: Scalarized and non-scalarized returns in the DRONE environment, with c=0.01. Uncertainty is parameterized by confidence level α and decreases left-to-right. Lines show the mean of 100 runs, with 95% confidence intervals shaded.

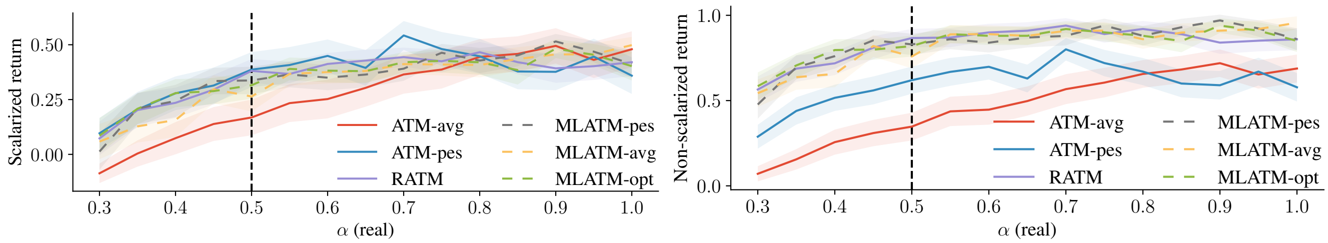

Figure 12: Scalarized and non-scalarized returns in the DRONE environment, with $c = 0.01$. Algorithms plan at uncertainty parametrized by $\alpha_p = 0.5$, but real dynamics is parametrized by α. Thus, real uncertainty decreases left-to-right, while planning uncertainty is given by the dotted line. Lines show the mean of 50 runs, with 95% confidence intervals shaded.

R(C)ATM outscales previous methods. RATM remains tractable and performs relatively well on the DRONE environment, which is not solvable by prior robust methods.

Measurement leniency can prevent conservative measuring. Our experiments in the LUCKY-UNLUCKY and SNAKEMAZE environment show measurement leniency incentivizes measuring in some uncertain settings. However, our experiments in the DRONE environment show this is not always the case. Thus, effectively using our methods may require looking at the structure of the model.

Related Work

Active-measure MDPs, with noiseless and complete measurements, were introduced independently by Nam, Fleming, and Brunskill (2021) and Bellinger et al. (2021), who both focussed on RL applications. Krale, Simão, and Jansen (2023) introduced the ATM heuristic, which finds a tradeoff between performance and scalability. Similar frameworks include those of Doshi-Velez, Pineau, and Roy (2012), who consider a setting where measurements return the optimal action, and Mate et al. (2020), who consider active measuring in a multi-armed bandit setting. Active measuring has also been considered in settings where measuring cost and rewards are not combined, but measuring costs are constrained (Ghasemi and Topcu 2019) or minimized (Bulychev et al. 2012). Lastly, some prior work considers setting with only measuring actions, where gathering information is the only goal. (Bernardino et al. 2022; Araya-López et al. 2011).

Although RAM-MDPs have not been studied previously, the more general framework of RPOMDPs has. Osogami (2015) and Cubuktepe et al. (2021) describe methods for solving adversarial RPOMDPs, using value iteration and finite state controllers, respectively. However, the former method scales poorly to large environments, while the latter only produces policies with small memory. Next, Rasouli and Saghafian (2018) gives an in-depth analysis of RPOMDPs with different assumptions, and Bovy (2023) describes how to represent uncertain beliefs without assuming adversariality. However, their methods are intractable for the sizes of environments considered here.

Conclusion

We introduced RAM-MDPs as a framework to represent active measuring environments with model uncertainty. To solve a specific subset of RAM-MDPs, we re-defined the act-then-measure heuristic for generic active-measure environments for uncertain settings. Next, we proposed *measurement leniency* to deal with overly conservative measuring behavior. We empirically evaluate both generic and measurement lenient variants of our algorithm, showing they are tractable and outperform non-robust baselines.

Future work will focus on making RATM more scalable, for example, by finding a way to approximately solve Equation (10), the current computational bottleneck. Moreover, we will explore more general (robust) active-measure environments with partial or noisy measurements.

Acknowledgments

This research has been partially funded by NWO grant NWA.1160.18.238 (PrimaVera) and the ERC Starting Grant 101077178 (DEUCE).

References

Araya-López, M.; Buffet, O.; Thomas, V.; and Charpillet, F. 2011. Active Learning of MDP Models. In *EWRL*, volume 7188 of *Lecture Notes in Computer Science*, 42–53. Springer.

Bellinger, C.; Coles, R.; Crowley, M.; and Tamblyn, I. 2021. Active Measure Reinforcement Learning for Observation Cost Minimization. In *Canadian Conference on AI*. Canadian Artificial Intelligence Association.

Bernardino, G.; Jonsson, A.; Loncaric, F.; Castellote, P. M.; Sitges, M.; Clarysse, P.; and Duchateau, N. 2022. Reinforcement Learning for Active Modality Selection During Diagnosis. In *MICCAI (1)*, volume 13431 of *Lecture Notes in Computer Science*, 592–601. Springer.

Bovy, E. 2023. The Underlying Belief Model of Uncertain Partially Observable Markov Decision Processes. *Master Thesis*.

Bulychev, P. E.; Cassez, F.; David, A.; Larsen, K. G.; Raskin, J.; and Reynier, P. 2012. Controllers with Minimal Observation Power (Application to Timed Systems). In *ATVA*, volume 7561 of *Lecture Notes in Computer Science*, 223–237. Springer.

Cubuktepe, M.; Jansen, N.; Junges, S.; Marandi, A.; Suilen, M.; and Topcu, U. 2021. Robust Finite-State Controllers for Uncertain POMDPs. In *AAAI*, 11792–11800. AAAI Press.

Doshi-Velez, F.; Pineau, J.; and Roy, N. 2012. Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs. *Artif. Intell.*, 187: 115–132.

Ghasemi, M.; and Topcu, U. 2019. Online Active Perception for Partially Observable Markov Decision Processes with Limited Budget. In *CDC*, 6169–6174. IEEE.

Jimenez-Roa, L. A.; Heskes, T.; Tinga, T.; Molegraaf, H. J.; and Stoelinga, M. 2022. Deterioration modeling of sewer pipes via discrete-time Markov chains: A large-scale case study in the Netherlands. In *32nd European Safety and Reliability Conference, ESREL 2022: Understanding and Managing Risk and Reliability for a Sustainable Future*, 1299–1306.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artif. Intell.*, 101(1-2): 99–134.

Kormushev, P.; Calinon, S.; and Caldwell, D. G. 2013. Reinforcement Learning in Robotics: Applications and Real-World Challenges. *Robotics*, 2(3): 122–148.

Krale, M.; Simão, T. D.; and Jansen, N. 2023. Act-Then-Measure: Reinforcement Learning for Partially Observable Environments with Active Measuring. In *ICAPS*, 212–220. AAAI Press.

Krale, M.; Simão, T. D.; Tumova, J.; and Jansen, N. 2023. Robust Active Measuring under Model Uncertainty. *arXiv preprint arXiv:2312.11227*.

Lei, L.; Tan, Y.; Zheng, K.; Liu, S.; Zhang, K.; and Shen, X. 2020. Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges. *IEEE Commun. Surv. Tutorials*, 22(3): 1722–1760.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning Policies for Partially Observable Environments: Scaling Up. In *ICML*, 362–370. Morgan Kaufmann.

Mate, A.; Killian, J. A.; Xu, H.; Perrault, A.; and Tambe, M. 2020. Collapsing Bandits and Their Application to Public Health Intervention. In *NeurIPS*.

Nam, H. A.; Fleming, S. L.; and Brunskill, E. 2021. Reinforcement Learning with State Observation Costs in Action-Contingent Noiselessly Observable Markov Decision Processes. In *NeurIPS*, 15650–15666.

Nilim, A.; and Ghaoui, L. E. 2005. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Oper. Res.*, 53(5): 780–798.

Osogami, T. 2012. Robustness and risk-sensitivity in Markov decision processes. In *NIPS*, 233–241.

Osogami, T. 2015. Robust partially observable Markov decision process. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 106–115. Lille, France: PMLR.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley.

Rasouli, M.; and Saghafian, S. 2018. Robust Partially Observable Markov Decision Processes. *SSRN Electronic Journal*.

Suilen, M.; Jansen, N.; Cubuktepe, M.; and Topcu, U. 2020. Robust Policy Synthesis for Uncertain POMDPs via Convex Optimization. In *IJCAI*, 4113–4120. ijcai.org.

Sunberg, Z.; and Kochenderfer, M. J. 2022. Improving Automated Driving Through POMDP Planning With Human Internal States. *IEEE Trans. Intell. Transp. Syst.*, 23(11): 20073–20083.

Wiesemann, W.; Kuhn, D.; and Rustem, B. 2013. Robust Markov Decision Processes. *Math. Oper. Res.*, 38(1): 153–183.

Yu, C.; Liu, J.; Nemati, S.; and Yin, G. 2023. Reinforcement Learning in Healthcare: A Survey. *ACM Comput. Surv.*, 55(2): 5:1–5:36.