Risk-Aware General-Utility Markov Decision Processes

Pedro P. Santos, Fábio Vital, Alberto Sardinha, Francisco S. Melo

Keywords: Risk-aware decision-making, Reinforcement learning, Planning.

Summary

We study general-utility Markov decision processes (GUMDPs) with risk-aware objectives. In this framework, an agent aims to optimize a risk measure of the distribution of objective values, where the objective function depends on the frequency of visitation of states induced by the agent's policy. First, we motivate, propose and formalize risk-aware GUMDPs, which enable agents and decision makers to trade off expected performance and risk aversion while benefiting from the rich set of objectives that can be cast under the framework of GUMDPs. We focus our attention to the entropic risk measure (ERM). Second, we show how we can solve risk-aware GUMDPs with ERM objectives by resorting to online planning techniques. In particular, we propose an MCTS-based approach to provably solve risk-aware GUMDPs up to any desired accuracy. Third, we provide a set of experimental results showcasing that our approach is successful when optimizing for a spectrum of risk-aware behaviors in the context of GUMDPs under diverse tasks (standard MDPs, maximum state entropy exploration, imitation learning, and multi-objective MDPs).

Contribution(s)

1. We motivate, propose and formalize risk-aware GUMDPs, which enable agents and decision makers to trade off expected performance and risk aversion while benefiting from the rich set of objectives that can be cast under the framework of GUMDPs.
   Context: Previous works studied risk-neutral (expected performance) policy optimization in GUMDPs (Zahavy et al., 2021; Mutti et al., 2023; Santos et al., 2025).
2. We show how we can solve risk-aware GUMDPs by resorting to online planning techniques, proposing an MCTS-based approach to provably solve risk-aware GUMDPs up to any desired accuracy.
   Context: None.
3. We provide a set of experimental results showcasing that our approach is successful when optimizing for a spectrum of risk-aware behaviors in the context of GUMDPs under diverse tasks (standard MDPs, maximum state entropy exploration, imitation learning, and multi-objective MDPs).
   Context: None.

Risk-Aware General-Utility Markov Decision Processes

Pedro P. Santos[1,2], **Fábio Vital**[1,2], **Alberto Sardinha**[1,3], **Francisco S. Melo**[1,2]
{pedro.pinto.santos, fabiovital}@tecnico.ulisboa.pt
sardinha@inf.puc-rio.br, fmelo@inesc-id.pt

[1]**INESC-ID**
[2]**Instituto Superior Técnico, University of Lisbon**
[3]**Pontifical Catholic University of Rio de Janeiro**

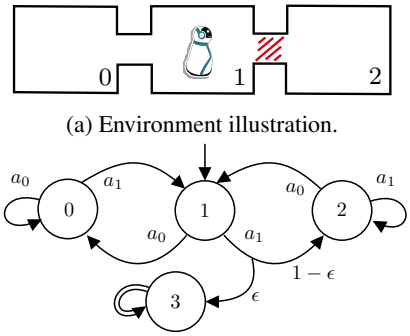
Abstract

We study general-utility Markov decision processes (GUMDPs) with risk-aware objectives. In this framework, an agent aims to optimize a risk measure of the distribution of objective values, where the objective function depends on the frequency of visitation of states induced by the agent's policy. First, we motivate, propose, and formalize risk-aware GUMDPs, which enable agents and decision makers to trade off expected performance by risk aversion while benefiting from the rich set of objectives that can be cast under the framework of GUMDPs. We focus our attention on the entropic risk measure (ERM). Second, we show how we can solve risk-aware GUMDPs with ERM objectives by resorting to online planning techniques. In particular, we propose an approach based on Monte Carlo Tree Search (MCTS) to provably solve risk-aware GUMDPs up to any desired accuracy. Third, we provide a set of experimental results showcasing that our approach is successful when optimizing for a spectrum of risk-aware behaviors in the context of GUMDPs under diverse tasks (standard MDPs, maximum state entropy exploration, imitation learning, and multi-objective MDPs). Code available at https://github.com/gh0stwin/risk-aware-gumdp.

1 Introduction

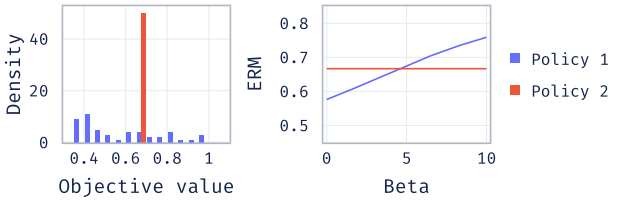
Markov decision processes (MDPs) (Puterman, 2014) have found a wide range of applications in different domains, ranging from optimal stopping (Chow et al., 1971), inventory management (Dvoretzky et al., 1952), or queueing control (Stidham, 1978). MDPs also play a central role in reinforcement learning (RL) (Sutton & Barto, 2018), where the agent-environment interaction is typically modelled within the MDP framework. In recent years, RL has achieved remarkable success across diverse domains (Mnih et al., 2015; Silver et al., 2017; Lillicrap et al., 2016).

However, many relevant objectives cannot be easily expressed within the standard MDP framework (Abel et al., 2022). Examples include imitation learning (Hussein et al., 2017; Osa et al., 2018), pure exploration problems (Hazan et al., 2019), risk-averse RL (García et al., 2015), diverse skills discovery (Eysenbach et al., 2018; Achiam et al., 2018), and constrained MDPs (Altman, 1999; Efroni et al., 2020). To address these limitations, the framework of general-utility Markov decision processes (GUMDPs) has been proposed as a more expressive formalism capable of modelling such objectives (Santos et al., 2024). In GUMDPs, the agent's objective is encoded as a function of the occupancy induced by a given policy, i.e., as a function of the frequency of visitation of state-action pairs induced by a given policy. GUMDPs generalize MDPs by allowing the objective function to be a non-linear function of occupancies.



(a) Environment illustration.

(b) Abstracted environment.



(c) Objective function density (left) and respective ERM$_\beta$ values (right) for two policies π_1 (Policy 1) and π_2 (Policy 2). Lower values are better.

Figure 1: Motivating example: Exploring three rooms.

Previous studies have been focused on studying policy optimization in GUMDPs by considering risk-neutral objectives, where the performance of a given policy is evaluated in terms of its induced *expected* frequency of visitation of states (Mutti et al., 2023; Santos et al., 2025; Zahavy et al., 2021; Geist et al., 2022). However, these approaches overlook the *variability* in the objective: for a fixed policy, the stochastic nature of the environment can lead to a distribution over possible state-action visitation frequencies, resulting in a range of potential objective values. Consequently, optimizing only for the expectation may fail to capture important aspects of the policy's performance. Given the stochasticity of the possible outcomes, it is thus natural to consider optimizing a specific *risk measure* (Artzner et al., 1999) over the distribution of objective values, enabling the learned behavior to be tuned toward risk-seeking or risk-averse preferences. The *entropic risk measure* (ERM) (Howard & Matheson, 1972) is one of the most common risk measures, which has received considerable attention across various domains such as finance and sequential decision-making (Föllmer & Schied, 2016; Borkar & Meyn, 2002; Pagnoncelli et al., 2022; Lin Hau et al., 2023; Marthe et al., 2025; Mortensen & Talebi, 2025). As a risk-aware criterion, the ERM enables agents/decision-makers to trade off between expected performance and risk aversion, providing robustness against adverse (worst-case) outcomes. To further motivate the relevance of trading off between risk-neutral and risk-averse behaviors in the context of GUMDPs, we present the following illustrative example.

Motivating example We consider the illustrative environment in Figure 1a, where a robot aims to explore three rooms as uniformly as possible (pure exploration task (Hazan et al., 2019)). Transitioning from room 1 to room 0 incurs no danger to the robot. However, attempting to transition from room 1 to room 2 may cause irreparable damage to the robot, rendering it unable to explore the environment further. The dynamics of this environment can be abstracted by the MDP in Figure 1b, where each state corresponds to one of the rooms, the actions encode the transitions between rooms, and $\epsilon \in (0, 1)$ is the probability of irreparable damage to the robot when attempting to transition between rooms 1 and 2. We can resort to a GUMDP to encode the agent's objective by setting the objective function to be (minus) the entropy of the occupancy induced by any policy. Now, a *risk-averse* agent will not be interested in trying to explore state 2 since there exists a probability that it will get absorbed into state 3 and, thus, not being able to explore the environment further. On the other hand, a *risk-seeker* agent will try to visit state 2 since it may be successful, hence exploring all three rooms. In Figure 1c, we compare two policies: (i) π_1, a *risk-seeking* policy that attempts to visit state 2; and (ii) π_2, a risk-averse policy that does not attempt to visit state 2 and, instead, keeps alternating between states 0 and 1. The left plot shows the density of the objective values for the policies. Both policies yield very different distributions, with the risk-seeking policy π_1 attaining lower (better) objective values at the cost of sometimes obtaining higher (worse) objective values. On the other hand, the risk-averse policy π_2 always attains the same intermediate objective value due to its deterministic behavior. The plot on the right shows the ERM$_\beta$ of both policies as a function of $\beta \in (0, \infty)$. Higher β values correspond to more risk-averse preferences, while β approaching

zero reflects risk neutrality. As shown, π_2 achieves a lower ERM_β value for higher β, confirming its risk-averse behavior. Conversely, as β decreases, π_1 becomes the superior policy.

Contributions We study risk-aware GUMDPs with ERM objectives. As a risk-aware criterion, the ERM enables agents/decision-makers to trade off between expected performance and risk aversion, providing robustness against adverse (worst-case) outcomes. Our *contributions* are threefold. *First*, we motivate, propose, and formalize risk-aware GUMDPs, which enable agents to trade off between expected performance and risk aversion while benefiting from the rich set of objectives that can be cast under the framework of GUMDPs. We focus our attention on the ERM, as it has been commonly used by previous works in the context of MDPs (Borkar & Meyn, 2002; Pagnoncelli et al., 2022; Lin Hau et al., 2023; Marthe et al., 2025; Mortensen & Talebi, 2025). *Second*, we show how we can solve risk-aware GUMDPs with ERM objectives (ERM-GUMDP) by resorting to online planning techniques. In particular, we show that any ERM-GUMDP can be reformulated as a specific MDP — referred to as an occupancy MDP — with an ERM objective. This reformulation enables the use of online planning techniques to compute approximately optimal policies for arbitrary ERM-GUMDPs. We propose an MCTS-based approach to provably solve risk-aware GUMDPs up to any desired accuracy. *Third*, we provide experimental results showing that our approach successfully optimizes for a spectrum of risk-aware behaviors in the context of GUMDPs under diverse tasks (standard MDPs, maximum state entropy exploration, imitation learning, and multi-objective MDPs).

2 Background

2.1 Markov decision processes

MDPs (Puterman, 2014) can be defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \{\boldsymbol{P}^a : a \in \mathcal{A}\}, \boldsymbol{p}_0, c)$ where: $\mathcal{S}$ is the discrete state space; $\mathcal{A}$ is the discrete action space; $\{\boldsymbol{P}^a : a \in \mathcal{A}\}$ is a set of transition probability matrices $\boldsymbol{P}^a$ for each action $a \in \mathcal{A}$; $\boldsymbol{p}_0 \in \Delta(\mathcal{S})$ is the distribution of initial states; and $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a cost function. The interaction between the agent and its environment, as modeled by the MDP framework, is as follows: (i) an initial state s_0 is sampled from $\boldsymbol{p}_0$; (ii) at each step t, the agent observes the state of the environment $\mathrm{s}_t \in \mathcal{S}$ and chooses an action $\mathrm{a}_t \in \mathcal{A}$. Depending on the action chosen by the agent, the MDP evolves to a new state $\mathrm{s}_{t+1} \in \mathcal{S}$ with probability given by $\boldsymbol{P}^{\mathrm{a}_t}(\mathrm{s}_t, \cdot)$, and the agent receives a cost $c(\mathrm{s}_t, \mathrm{a}_t)$; and (iii) the interaction repeats infinitely.

A decision rule π_t specifies the action-selection mechanism at timestep t. A non-Markovian decision rule maps the history of states and actions to a probability distribution over actions, i.e., $\pi_t : \mathcal{S} \times (\mathcal{S} \times \mathcal{A})^t \to \Delta(\mathcal{A})$. In contrast, a Markovian decision rule depends only on the current state and maps it to a distribution over actions, i.e., $\pi_t : \mathcal{S} \to \Delta(\mathcal{A})$. Both types of decision rules can be deterministic if they map directly to actions rather than distributions: for non-Markovian rules, $\pi_t : \mathcal{S} \times (\mathcal{S} \times \mathcal{A})^t \to \mathcal{A}$, and for Markovian rules, $\pi_t : \mathcal{S} \to \mathcal{A}$. A policy $\pi = (\pi_0, \pi_1, \ldots)$ is a sequence of decision rules, one for each timestep. A policy is Markovian or non-Markovian if all its decision rules are Markovian or non-Markovian, respectively. Likewise, a policy is deterministic or stochastic if all its decision rules are deterministic or stochastic. We denote the set of non-Markovian policies by Π_{NM}, Markovian policies by Π_{M}, non-Markovian deterministic policies by $\Pi^{\text{D}}_{\text{NM}}$, and Markovian deterministic policies by $\Pi^{\text{D}}_{\text{M}}$. A policy is stationary if it uses the same decision rule at every timestep. We denote the set of stationary policies by Π_{S} and the set of stationary deterministic policies by $\Pi^{\text{D}}_{\text{S}}$. Among the relationships between these classes, we highlight the following: (i) Π_{NM} is the most general class; (ii) $\Pi_{\text{S}} \subset \Pi_{\text{M}} \subset \Pi_{\text{NM}}$; and (iii) $\Pi^{\text{D}}_{\text{S}} \subset \Pi^{\text{D}}_{\text{M}} \subset \Pi^{\text{D}}_{\text{NM}}$.

For a given policy $\pi \in \Pi_{\text{NM}}$, the interaction between the agent and the environment induces a random process $(\mathrm{s}_0, \mathrm{a}_0, \mathrm{s}_1, \mathrm{a}_1, \ldots)$. We denote by $\mathrm{h}_t = (\mathrm{s}_0, \mathrm{a}_0, \mathrm{s}_1, \mathrm{a}_1, \ldots, \mathrm{s}_t)$ the random history up to and including timestep t, and by $h_t = (s_0, a_0, s_1, a_1, \ldots, s_t) \in \mathcal{S} \times (\mathcal{S} \times \mathcal{A})^t$ a particular realization of such a history. The random process $(\mathrm{s}_0, \mathrm{a}_0, \mathrm{s}_1, \mathrm{a}_1, \ldots)$ satisfies the following conditions: (i) $\mathbb{P}[\mathrm{s}_0 = s] = p_0(s)$; (ii) $\mathbb{P}[\mathrm{s}_{t+1} = s' \mid \mathrm{h}_t, \mathrm{a}_t] = P^{\mathrm{a}_t}(\mathrm{s}_t, s')$; and (iii) $\mathbb{P}[\mathrm{a}_t = a \mid \mathrm{h}_t] = \pi(a \mid \mathrm{h}_t)$. Let $(\Omega, \mathcal{F}, \mathbb{P}_\pi)$ be the probability space over trajectories $(\mathrm{s}_0, \mathrm{a}_0, \mathrm{s}_1, \mathrm{a}_1, \ldots)$ satisfying (i)–(iii), as defined in Lattimore & Szepesvári (2020). We denote a sample trajectory by $\omega \in \Omega$, where

$\omega = (s_0, a_0, s_1, a_1, \ldots)$. We also let $\mathbb{P}_\pi[\mathrm{s}_t = s, \mathrm{a}_t = a]$ be the probability of observing the state-action pair (s, a) at timestep t under the MDP while following policy π.

The expected discounted cumulative cost objective is $J^\pi_\gamma = \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t c(\mathrm{s}_t, \mathrm{a}_t)\right]$, where $\gamma \in (0, 1)$ is the discount factor and the expectation is taken over the random trajectory of state-action pairs. The discounted state-action occupancy for policy π is defined as

$$d_\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\pi \left[\mathrm{s}_t = s, \mathrm{a}_t = a\right]. \tag{1}$$

The expected discounted cumulative cost of policy π can be rewritten as $J^\pi_\gamma = \boldsymbol{c}^\top \boldsymbol{d}_\pi$, where $\boldsymbol{d}_\pi = [d_\pi(s_0, a_0), \ldots, d_\pi(s_{|\mathcal{S}|}, a_{|\mathcal{A}|})]^\top$, $\boldsymbol{c} = [c(s_0, a_0), \ldots, c(s_{|\mathcal{S}|}, a_{|\mathcal{A}|})]^\top$. We aim to find $\pi^* = \arg\min_{\pi \in \Pi_\mathrm{s}} J^\pi_\gamma$, which can be formulated as a linear program (Puterman, 2014).

2.2 Risk-aware MDPs

Risk-aware MDPs (Chow et al., 2015) generalize the classical MDP framework by considering the associated variability of the discounted cumulative costs. This is done by considering $J^\pi_{\gamma,\rho} = \rho\left(\sum_{t=0}^\infty \gamma^t c(\mathrm{s}_t, \mathrm{a}_t)\right)$, where ρ is a *risk measure* (Artzner et al., 1999), computed over the random trajectory. Risk measures quantify an agent's attitude toward uncertainty and variability in outcomes. We note that for $\rho(\cdot) = \mathbb{E}[\cdot]$ we have the standard discounted MDP objective. The risk-aware objective allows the agent to reason over the entire distribution instead of just its expectation.

In this work, we consider risk-aware MDPs with an ERM objective (Howard & Matheson, 1972). The ERM of a random variable z is defined as $\mathrm{ERM}_\beta(\mathrm{z}) = \frac{1}{\beta} \ln\left(\mathbb{E}\left[\exp(\beta \mathrm{z})\right]\right)$, where parameter $\beta \in (0, \infty)$ trades off between expected performance as $\beta \to 0$ and risk-aversion as β increases. Also, the ERM_β belongs to the family of optimized certainty equivalents (OCEs) (Ben-Tal & Teboulle, 1986) since it can be equivalently defined as $\mathrm{ERM}_\beta(\mathrm{z}) = \min_{\lambda \in \mathbb{R}} \left\{\lambda + \mathbb{E}\left[u(\mathrm{z} - \lambda)\right]\right\}$, where $u(x) = \frac{1}{\beta}\left(\exp(\beta x) - 1\right)$. In this work, we focus our attention on the ERM_β and, thus, we implicitly refer to the ERM_β whenever we write ρ.

2.3 General-utility MDPs

GUMDPs generalize utility specification by allowing the agent's objective to be written in terms of the frequency of visitation of state-action pairs. This is in contrast to the standard MDPs framework, where the objective of the agent is encoded by the cost function.

We define an infinite-horizon discounted GUMDP as a tuple $\mathcal{M}_f = (\mathcal{S}, \mathcal{A}, \{\boldsymbol{P}^a : a \in \mathcal{A}\}, \boldsymbol{p}_0, f)$, where $\mathcal{S}$, $\mathcal{A}$, $\{\boldsymbol{P}^a : a \in \mathcal{A}\}$, and $\boldsymbol{p}_0$ are defined in a similar way to the standard MDP formulation, as introduced in Sec. 2.1. The function $f : \Delta(\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$ encodes the objective of the agent, which depends on a discounted occupancy $\boldsymbol{d}$, as defined in (1). We then aim to find $\pi^* \in \arg\min_{\pi \in \Pi_\mathrm{s}} f(\boldsymbol{d}_\pi)$. If f is a linear function, then we are under the standard MDP setting. If f is convex, then we are under the convex MDP/RL setting (Zahavy et al., 2021).

In this work, we consider four tasks, each associated with a particular objective function: (i) cost minimization (standard MDP) (Puterman, 2014), where $f(\boldsymbol{d}) = \boldsymbol{c}^\top \boldsymbol{d}$ and $\boldsymbol{c} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is the column vector encoding the cost function; (ii) maximum state entropy exploration (MSEE) (Hazan et al., 2019), where $f(\boldsymbol{d}) = \boldsymbol{d}^\top \log(\boldsymbol{d})$, i.e., we aim to find a policy that visits all state-action pairs as uniformly as possible; (iii) imitation learning (IL) (Abbeel & Ng, 2004), where $f(\boldsymbol{d}) = \|\boldsymbol{d} - \boldsymbol{d}_{\pi_b}\|_2^2$ with $\boldsymbol{d}_{\pi_b} \in \Delta(\mathcal{S} \times \mathcal{A})$, i.e., we aim to find a policy such that its induced occupancy is as similar as possible to the occupancy $\boldsymbol{d}_{\pi_b}$ induced by a given behavior policy π_b; and (iv) multi-objective (MO) MDPs, where $f(d) = g(\boldsymbol{d}^\top \boldsymbol{c}_1, \ldots, \boldsymbol{d}^\top \boldsymbol{c}_k)$, where $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ are vectorized cost functions and g is an utility function (Radulescu et al., 2019). Nevertheless, we highlight that our results apply to any task that can be modeled using the GUMDPs framework. We refer to Zahavy et al. (2021) for a comprehensive list of the different objectives considered by previous works.

3 Towards Risk-Aware GUMDPs

We now introduce the framework of risk-aware GUMDPs. Because risk-awareness requires reasoning about the underlying distribution of objective values, we begin by adopting a distributional perspective on GUMDPs. Similarly to distributional RL (Bellemare et al., 2023), we employ the following assumption to ensure that all random variables defined below are bounded and well-defined.

Assumption 3.1. The objective f satisfies $f(\boldsymbol{d}) \in \mathcal{Y} = [-R, R], \forall \boldsymbol{d} \in \Delta(\mathcal{S} \times \mathcal{A})$, where $R \in \mathbb{R}^+$.

3.1 A distributional perspective on GUMDPs

For a given fixed policy $\pi \in \Pi_{\text{NM}}$, let $\mathbf{d}^\pi : \Omega \to \Delta(\mathcal{S} \times \mathcal{A})$ be the random vector defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P}_\pi)$ with entries

$$\mathrm{d}^\pi_{s,a}(\omega) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbf{1}(s_t = s, a_t = a). \tag{2}$$

The random vector above is well-defined since the mapping $\mathbf{d}^\pi$ is $\mathcal{F}/\mathcal{B}(\Delta(\mathcal{S} \times \mathcal{A}))$ measurable, where $\mathcal{B}(\Delta(\mathcal{S} \times \mathcal{A}))$ denotes the Borel σ-algebra of $\Delta(\mathcal{S} \times \mathcal{A})$. Let also $\mathrm{f}^\pi : \Omega \to \mathcal{Y}$ be the random variable with support in $\mathcal{Y}$ such that

$$\mathrm{f}^\pi(\omega) = (f \circ \mathbf{d}^\pi)(\omega) = f(\mathbf{d}^\pi(\omega)),$$

satisfying

$$\mathbb{P}\left[\mathrm{f}^\pi \leq \eta\right] = \mathbb{P}_\pi\left[\{\omega \in \Omega : f(\mathbf{d}^\pi(\omega)) \leq \eta\}\right], \quad \forall \eta \in \mathcal{Y},$$

where we assume f is a Borel measurable function. Essentially, the random variable f^π is defined as the composition of two measurable mappings. If f is continuous, then f is Borel measurable.

We refer to Figure 1c for an illustration of the density associated with the random variable f^π for two arbitrary policies. As observed, the underlying distributions induced by different policies can differ substantially and, therefore, depending on the exact preferences of the decision-maker, one policy may be preferable over the other. Risk measures, such as the ERM_β, allow for the comparison between distributions induced by different policies.

3.2 Risk-aware GUMDPs

In the context of GUMDPs, we denote $\mathcal{L}$ the set of possible random variables with support in $\mathcal{Y}$ associated with each of the probability spaces $(\Omega, \mathcal{F}, \mathbb{P}_\pi)$, for all $\pi \in \Pi_{\text{NM}}$. Note that $\mathrm{f}^\pi \in \mathcal{L}$ for any $\pi \in \Pi_{\text{NM}}$. Then, we consider risk measures $\rho : \mathcal{L} \to \mathbb{R}$ that associate any random variable in $\mathcal{L}$ with a scalar value. In this work, we are particularly interested in the ERM_β, as introduced in Sec. 2.2. We aim to find

$$\pi^* = \arg\min_{\pi \in \Pi_{\text{NM}}} \rho(\mathrm{f}^\pi). \tag{3}$$

For any policy $\pi \in \Pi_{\text{NM}}$, we define its optimality gap as $\text{OptGap}(\pi) = \rho\left(\mathrm{f}^\pi\right) - \min_{\pi' \in \Pi_{\text{NM}}} \rho\left(\mathrm{f}^{\pi'}\right)$. Essentially, $\text{OptGap}(\pi)$ measures how suboptimal a given policy π is compared to the best policy.

Policy optimization in risk-aware GUMDPs While in the context of risk-aware MDPs with ERM objectives the class of deterministic and Markovian policies, $\Pi^{\text{D}}_{\text{M}}$, suffices for optimality (Lin Hau et al., 2023), this is no longer the case in the context of risk-aware GUMDPs with ERM objectives. As shown in Santos et al. (2025) in the context of GUMDPs, if $\rho(\cdot) = \mathbb{E}[\cdot]$ then the class of Markovian policies is strictly dominated by the class of non-Markovian policies. Since, in the context of risk-aware GUMDPs, as $\beta \to 0$ the ERM_β approaches $\mathbb{E}[\cdot]$, non-Markovian policies are also needed, in general, in the context of solving GUMDPs with ERM objectives.

4 Solving Risk-Aware GUMDPs with Online Planning

In this section, we study how to solve risk-aware GUMDPs using online planning techniques. We begin by establishing a close connection between risk-aware GUMDPs and a particular risk-aware MDP in which the agent tracks the accrued occupancy at each timestep. Leveraging this formulation, we show that solving the risk-aware MDP with online planning techniques yields approximately optimal action selection for the original risk-aware GUMDP.

Assumption 4.1. The objective function f is L_f-Lipschitz, i.e., $|f(\boldsymbol{d}_1) - f(\boldsymbol{d}_2)| \leq L_f \|\boldsymbol{d}_1 - \boldsymbol{d}_2\|_1$, for any $\boldsymbol{d}_1, \boldsymbol{d}_2 \in \Delta(\mathcal{S} \times \mathcal{A})$.

We refer to Appendix 8 for the derivation of L_f-Lipschitz constants for the objective functions considered.

4.1 Computing optimal policies by resorting to finite-horizon GUMDPs

We start by showing that we can compute an approximately optimal policy for our infinite-horizon discounted risk-aware objective (3) by resorting to a finite-horizon formulation of the same problem. For any policy $\pi \in \Pi_{\text{NM}}$, let $\mathrm{f}_H^\pi : \Omega \to \mathcal{Y}$ be the random variable with support in $\mathcal{Y}$ such that

$$\mathrm{f}_H^\pi(\omega) = (f \circ \mathbf{d}^{\pi,H})(\omega) = f(\mathbf{d}^{\pi,H}(\omega)) \tag{4}$$

$$\mathrm{d}_{s,a}^{\pi,H}(\omega) = \frac{1-\gamma}{1-\gamma^H} \sum_{t=0}^{H-1} \gamma^t \mathbf{1}(s_t = s, a_t = a). \tag{5}$$

Essentially, random variable f_H^π is defined in a similar fashion to the random variable f^π, but the former considers a random occupancy $\mathbf{d}^{\pi,H}$ that only takes into account the state-action pairs visited up to a given horizon $H \in \mathbb{N}$. We state the following result (proof in Appendix 9).

Proposition 4.2 (Optimality gap decomposition)**.** *For arbitrary $\pi \in \Pi_{\text{NM}}$, it holds that*

$$\mathrm{OptGap}(\pi) \leq \underbrace{\rho(\mathrm{f}_H^\pi) - \min_{\pi' \in \Pi_{NM}} \left\{ \rho(\mathrm{f}_H^{\pi'}) \right\}}_{= \mathrm{OptGap}_H(\pi)} + 8 L_f \gamma^H,$$

where $\mathrm{OptGap}_H(\pi)$ is the optimality gap of policy π for the truncated objective with horizon H.

The result above shows that we can resort to the truncated objective to compute approximately optimal policies, up to any desired accuracy, for the infinite-horizon risk-aware objective (3). Therefore, we focus our attention on solving $\pi^* = \arg\min_{\pi \in \Pi_{\text{NM}}} \rho(\mathrm{f}_H^\pi)$, where f_H^π is defined in (4). Since we now focus on the case of finite-horizon trajectories, we assume that, for a given policy $\pi \in \Pi_{\text{NM}}$, the interaction between the agent and the environment gives rise to a random process $(\mathrm{s}_0, \mathrm{a}_0, \mathrm{s}_1, \mathrm{a}_1, \ldots, \mathrm{s}_{H-1}, \mathrm{a}_{H-1})$ such that: (i) $\mathbb{P}[\mathrm{s}_0 = s] = p_0(s)$; (ii) $\mathbb{P}[\mathrm{s}_{t+1} = s' | \mathrm{h}_t, \mathrm{a}_t] = P^{\mathrm{a}_t}(\mathrm{s}_t, s')$; and (iii) $\mathbb{P}[\mathrm{a}_t = a | \mathrm{h}_t] = \pi(a|\mathrm{h}_t)$. We thus let from now on $(\Omega, \mathcal{F}, \mathbb{P}_\pi)$ be the probability space over the sequence of random variables $(\mathrm{s}_0, \mathrm{a}_0, \mathrm{s}_1, \mathrm{a}_1, \ldots, \mathrm{s}_{H-1}, \mathrm{a}_{H-1})$ that satisfies conditions (i)-(iii) above. We write specific trajectories as $\omega \in \Omega$, with $\omega = (s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1})$. We highlight that the probability of a given trajectory $\omega \in \Omega$ under policy $\pi \in \Pi_{\text{NM}}$ can be calculated as $\mathbb{P}_\pi[\omega] = p_0(s_0) \cdot \pi(a_0|h_0) \cdot P^{a_0}(s_0, s_1) \cdot \pi(a_1|h_1) \cdot P^{a_1}(s_1, s_2) \ldots P^{a_{H-2}}(s_{H-2}, s_{H-1}) \cdot \pi(a_{H-1}|h_{H-1})$.

4.2 The occupancy MDP

To derive our planning algorithms for solving risk-aware GUMDPs, we make use of a finite-horizon MDP derived from the original GUMDP formulation. In particular, we consider the occupancy MDP (Santos et al., 2025), defined by the tuple $\mathcal{M}_{\text{O}} = \{\mathcal{S}_{\text{O}}, \mathcal{A}_{\text{O}}, \{\boldsymbol{P}_{\text{O}}^a : a \in \mathcal{A}\}, \boldsymbol{p}_{0,\text{O}}, c_{\text{O}}, H\}$, where $\mathcal{S}_{\text{O}} = \mathcal{S} \times \mathcal{O}$ is the discrete state space and $\mathcal{O}$ is the set of occupancies up to length $H-1$ in the original GUMDP. We let $\{s, \boldsymbol{o}\}$ be a state of the occupancy MDP such that $s \in \mathcal{S}$ is a state

from the original GUMDP and $\boldsymbol{o} \in \mathcal{O}$ is a $|\mathcal{S}||\mathcal{A}|$-dimensional vector that keeps track of the running occupancy of the agent up to a given timestep. Then, $\mathcal{A}_{\mathrm{O}} = \mathcal{A}$ is the action space and $\boldsymbol{p}_{0,\mathrm{O}}$ is such that $p_{0,\mathrm{O}}(\{s, \boldsymbol{o}\}) = p_0(s)$ if $\boldsymbol{o} = [0, \ldots, 0]$ and zero otherwise. The dynamics are as follows: (i) component $\mathrm{s}_{t+1} \sim \boldsymbol{P}^{\mathrm{a}_t}(\cdot|\mathrm{s}_t)$ evolves according to the dynamics of the original GUMDP; and (ii) the running occupancy evolves deterministically as $o_{t+1}(s, a) = \gamma^t + o_t(s, a)$ if $s = \mathrm{s}_t$ and $a = \mathrm{a}_t$, and $o_{t+1}(s, a) = o_t(s, a)$ otherwise. Finally, $H \in \mathbb{N}$ denotes the horizon and the cost function $c_{\mathrm{O}} : \mathcal{S} \times \mathcal{O} \to \mathbb{R}$ is defined as

$$c_{\mathrm{O}}(\{s, \boldsymbol{o}\}) = \begin{cases} 0 & \text{if } t < H, \\ f\left(\frac{1-\gamma}{1-\gamma^H}\boldsymbol{o}\right) & \text{if } t = H. \end{cases}$$

Stationary policies $\pi_{\mathrm{O}} \in \Pi_{\mathrm{S}}$ for $\mathcal{M}_{\mathrm{O}}$ are mappings of the type $\pi_{\mathrm{O}} : \mathcal{S} \times \mathcal{O} \to \Delta(\mathcal{A})$. For a given policy $\pi_{\mathrm{O}} \in \Pi_{\mathrm{S}}$, the interaction between the agent and the occupancy MDP gives rise to a random process $(\{\mathrm{s}_0, \mathbf{o}_0\}, \mathrm{a}_0, \{\mathrm{s}_1, \mathbf{o}_1\}, \mathrm{a}_1, \ldots, \{\mathrm{s}_H, \mathbf{o}_H\})$ associated with the probability space $(\Omega_{\mathrm{O}}, \mathcal{F}_{\mathrm{O}}, \mathbb{P}^{O}_{\pi_{\mathrm{O}}})$. We write specific trajectories as $\omega_{\mathrm{O}} \in \Omega_{\mathrm{O}}$, with $\omega_{\mathrm{O}} = (\{s_0, \boldsymbol{o}_0\}, a_0, \{s_1, \boldsymbol{o}_1\}, a_1, \ldots, \{s_H, \boldsymbol{o}_H\})$.

For any policy $\pi_{\mathrm{O}} \in \Pi_{\mathrm{S}}$, we let $\mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}} : \Omega_{\mathrm{O}} \to \mathcal{Y}$ be the random variable with support in $\mathcal{Y}$ associated with the probability space $(\Omega_{\mathrm{O}}, \mathcal{F}_{\mathrm{O}}, \mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}})$ such that $\mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}}(\omega_{\mathrm{O}}) = \sum_{t=0}^{H} c_{\mathrm{O}}(\{s_t, \boldsymbol{o}_t\})$. We reproduce the following result from Santos et al. (2025).

Lemma 4.3 (One-to-one mapping between histories in $\mathcal{M}_f$ and states in $\mathcal{M}_{\mathrm{O}}$)**.** *There exists a one-to-one mapping between histories $h_l = (s_0, a_0, s_1, a_1, \ldots, s_l) \in \mathcal{S} \times (\mathcal{S} \times \mathcal{A})^l$ in $\mathcal{M}_f$, with $0 \leq l \leq H - 1$, and states $\{s, \boldsymbol{o}\} \in \mathcal{S} \times \mathcal{O}$ in $\mathcal{M}_O$.*

An important conclusion that can be derived from the result above is that there exists a one-to-one mapping between non-Markovian policies for $\mathcal{M}_f$ and stationary policies for $\mathcal{M}_{\mathrm{O}}$. This is because every state in $\mathcal{M}_{\mathrm{O}}$ is uniquely associated with a particular history in $\mathcal{M}_f$ (and vice versa), as the result above shows. With this in mind, we now state the following results (proofs in Appendix 9).

Lemma 4.4 (Equivalence in distribution between f^{π}_{H} and $\mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}}$)**.** *For any horizon $H \in \mathbb{N}$ and policy $\pi \in \Pi_{NM}$, it holds that $\mathrm{f}^{\pi}_{H} \overset{D}{=} \mathrm{J}^{\pi_O}_{O}$, i.e., random variables f^{π}_{H} and $\mathrm{J}^{\pi_O}_{O}$ are equal in distribution, where π_O is the stationary policy for $\mathcal{M}_O$ associated with the non-Markovian policy π for $\mathcal{M}_f$.*

Theorem 4.5 (Solving the risk-aware $\mathcal{M}_f$ is "equivalent" to solving the risk-aware $\mathcal{M}_{\mathrm{O}}$)**.** *For a given risk measure $\rho : \mathcal{L} \to \mathbb{R}$, the problem of finding a policy $\pi \in \Pi_{NM}$ satisfying $\mathrm{OptGap}_H(\pi) \leq \epsilon$, for any $\epsilon \in \mathbb{R}^+_0$, can be reduced to the problem of finding a policy $\pi_O \in \Pi_S$ satisfying*

$$\rho\left(\mathrm{J}^{\pi_O}_{O}\right) - \min_{\pi'_O \in \Pi_S} \rho\left(\mathrm{J}^{\pi'_O}_{O}\right) \leq \epsilon.$$

In particular, if $\pi^*_O = \arg\min_{\pi_O \in \Pi_S} \rho(\mathrm{J}^{\pi_O}_{O})$, then the corresponding non-Markovian policy π in $\mathcal{M}_f$ satisfies $\mathrm{OptGap}_H(\pi) = 0$. It also holds that $\mathrm{OptGap}_H(\pi) = \rho\left(\mathrm{J}^{\pi_O}_{O}\right) - \min_{\pi'_O \in \Pi_S} \rho\left(\mathrm{J}^{\pi'_O}_{O}\right)$, where π_O is the stationary policy for $\mathcal{M}_O$ associated with the non-Markovian policy π for $\mathcal{M}_f$.

The result above shows that it suffices to search for an (approximately) stationary optimal policy for $\mathcal{M}_{\mathrm{O}}$, since such a policy corresponds to a non-Markovian policy that is (approximately) optimal for $\mathcal{M}_f$. In particular, such an (approximately) optimal policy for $\mathcal{M}_{\mathrm{O}}$ can be seen as a non-Markovian policy for $\mathcal{M}_f$ that compresses the history up to any timestep into a running occupancy. The result above shows that *it suffices to keep track of the running occupancy up to any timestep in order to attain optimal behavior* when solving risk-aware GUMDPs.

In light of Theorem 4.5, we consider risk-aware planning algorithms to solve the occupancy MDP. Unfortunately, solving the risk-aware occupancy MDP poses some challenges. One of the key challenges is due to the fact that the size of the state space of the occupancy MDP grows combinatorially with H since every state in the occupancy MDP is associated with a possible history in $\mathcal{M}_f$. Consequently, the state space of the occupancy MDP is typically very large, which precludes the use of offline planning methods. In the next section, we therefore turn to online planning approaches to implicitly compute approximately risk-aware optimal policies for the occupancy MDP.

4.3 Solving risk-aware occupancy MDPs via online planning

We aim to propose a practical algorithm to compute an approximately optimal risk-aware policy for the occupancy MDP $\mathcal{M}_O$, as introduced in Sec. 4.2. Since the occupancy MDP is a standard undiscounted finite-horizon MDP, we can resort to the risk-aware Monte Carlo tree search (MCTS) algorithm put forth by Santos et al. (2026), ERM-MCTS, to solve the risk-aware occupancy MDP with the ERM objective. ERM-MCTS works in a similar fashion to standard MCTS in the sense that the algorithm iteratively builds a search tree that alternates between decision nodes, where actions are selected, and chance nodes corresponding to random next states sampled from the MDP. At each iteration, ERM-MCTS refines the search tree by simulating a random trajectory in the occupancy MDP. Action-selection at each decision node $\mathrm{s}_t \in \mathcal{S}_O$ is given by

$$\mathrm{a}_t \in \arg\min_{a \in \mathcal{A}} \left\{ \frac{1}{\beta} \ln \left(\frac{1}{N(\mathrm{s}_t, a)} \sum_{i=1}^{N(\mathrm{s}_t, a)} \exp\left(\beta \mathrm{x}_i^{(\mathrm{s}_t, a)}\right) \right) - \theta_t \sqrt{\frac{\sqrt{N(\mathrm{s}_t)}}{N(\mathrm{s}_t, a)}} \right\},$$

where $N(\mathrm{s}_t)$ is the number of times s_t has been visited, $N(\mathrm{s}_t, a)$ the number of times action a has been selected while in s_t, $\mathrm{x}_i^{(\mathrm{s}_t, a)}$ is the i-th sampled terminal cost starting from (s_t, a) (given that the occupancy MDP only has non-zero costs at $t = H$), and θ_t is an exploration constant. In case $N(\mathrm{s}_t, a) = 0$ for some $a \in \mathcal{A}$, then a is selected. We refer to Appendix 10 for the full pseudocode of the ERM-MCTS algorithm. Without loss of generality, assume we fix an initial state $s_0 \in \mathcal{S}$. We state the following result, which is a consequence of Theo. 6 in Santos et al. (2026).

Theorem 4.6. *There exist exploration constants $(\theta_0, \theta_1, \ldots, \theta_{H-1})$ such that, from any initial state $s_0 \in \mathcal{S}$,* ERM-MCTS *provably solves the risk-aware occupancy MDP with an ERM objective. More precisely, let $\hat{V}_n(s_0) = \frac{1}{\beta} \ln \left(\frac{1}{n} \sum_{a \in \mathcal{A}} \sum_{i=1}^{T_a(n)} \exp\left(\beta \mathrm{x}_i^{(\mathrm{s}_t, a)}\right) \right)$ be the empirical ERM obtained by the* ERM-MCTS *algorithm after n iterations at initial state s_0, where $T_a(n)$ denotes the number of times action a was selected at root node s_0 after n iterations. Then,*

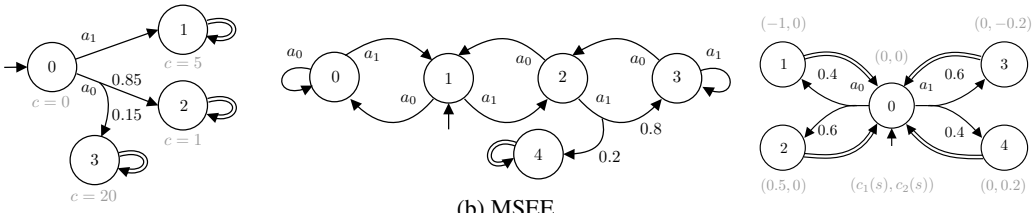
$$\lim_{n \to \infty} \mathbb{E}\left[\hat{V}_n(s_0)\right] \overset{(a)}{=} \min_{\pi'_O \in \Pi_S} \rho\left(\mathrm{J}_O^{\pi'_O}\right) \overset{(b)}{=} \min_{\pi' \in \Pi_{NM}} \left\{ \rho(\mathrm{f}_H^{\pi'}) \right\},$$

where: (a) follows from Theo. 6 in Santos et al. (2026) as the expected empirical ERM obtained by ERM-MCTS *converges, in the limit, to the optimal ERM for the occupancy MDP; and (b) follows from Theo. 4.5.*

5 Experimental Results

We empirically assess the performance of ERM-MCTS for solving risk-aware GUMDPs, investigating how ERM-MCTS trades off risk-neutral and risk-averse behavior. We describe our experimental methodology and refer to Appendix 11 for a complete description of our experiments (environments, baselines, hyperparameters, etc.).

We consider four tasks: (i) cost minimization (MDP); (ii) maximum state entropy exploration (MSEE); (iii) imitation learning (IL); and (iv) multi-objective (MO) MDPs with different utility functions. The definition of the objective function for each task is given in Sec. 2.3. We consider two sets of environments: (i) illustrative environments that consist of low-dimensional GUMDPs; and (ii) high-dimensional GUMDPs consisting of grid-based environments showcasing different sources of stochasticity in order to encourage distinct types of policy behaviors. To our knowledge, we are the first work to propose an algorithm to solve risk-aware GUMDPs and, hence, there are no baselines that we can use to compare the performance of our method against. However, for the particular case of linear f (MDP), we use an oracle baseline ERM-BI to validate the performance of our ERM-MCTS algorithm. ERM-BI computes the optimal risk-aware policy for the underlying occupancy MDP using a dynamic programming approach by exploiting the dynamic decomposition of the ERM_β put forth by Lin Hau et al. (2023). All our plots are computed by aggregating the experimental results of, at least, 100 independent runs, and we refer to Tables 2 and 3 for the detailed list of our experimental hyperparameters across all environments.



(a) MDP and IL (10% reset)

(b) MSEE

(c) MO

Figure 2: Illustrative environments.

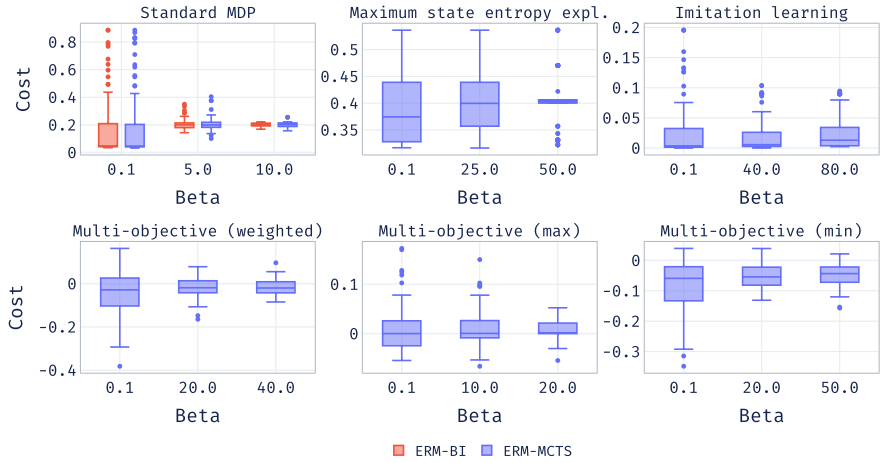


Figure 3: Box plots of the costs obtained under the illustrative environments. Lower is better.

5.1 Illustrative environments

We display in Fig. 2 our illustrative environments. The MDP (Fig. 2 (a)) consists of a four-state MDP where the agent needs to tradeoff at the initial state (s_0) between: (i) a risky action (a_0) that can lead to a low-cost state (s_2), but with some probability the agent ends in a high-cost state (s_3); and (ii) a safe action (a_1) that deterministically leads to a medium-cost state (s_1). The agent resets to the initial state with 10% probability while not in the initial state. For the IL task (Fig. 2 (a)), the agent aims to imitate the empirical occupancy induced by the trajectory of an agent that selected twice the risky action (a_0) and then selected the safe action (a_1) for all the remainder timesteps. The trajectory to imitate had a rather "lucky" outcome, as it never ended up in the absorbing state (s_3). Hence, the agent needs to trade off between imitating the behavior policy in states (s_0, s_1, s_2) and risking being absorbed into s_3, or only imitating the behavior policy in the "less risky" states (s_0, s_1). The MSEE task (Fig. 2 (b)) is similar to the motivating example from Sec. 1, where the agent aims to explore an environment as uniformly as possible, but there is a chance that the agent transitions to an absorbing state (s_4). Finally, the MO-MDP (Fig. 2 (c)) is inspired by the FishWood environment (Roijers et al., 2020), where the agent needs to tradeoff between two cost functions. Each cost function penalizes different behaviors. We consider three utility functions (Hayes et al., 2022): (i) a weighted combination of the discounted cumulative costs; (ii) the maximum of the discounted cumulative costs; and (iii) the minimum of the discounted cumulative costs.

In Fig. 3, we display the box plots computed for the empirical distributions of costs obtained by ERM-MCTS under different tasks and β values. To validate our algorithm, we include a cost-minimization task corresponding to a standard MDP. As seen, for the MDP, the box plot obtained by ERM-MCTS closely matches that obtained by the ERM-BI oracle baseline across the tested β

values, providing empirical validation of ERM-MCTS. Furthermore, across all tasks, ERM-MCTS successfully trades off risk-neutral and risk-averse behavior as a function of β. As β increases and ERM-MCTS progressively focuses on optimizing worst-case outcomes, upper whiskers and outliers (i.e., the worst outcomes) of the box plots decrease. This indicates that worst-case outcomes become less probable and that ERM-MCTS effectively computes increasingly risk-averse policies. Naturally, optimizing for worst-case outcomes typically degrades best-case or expected outcomes, as reflected by the increase in median values and in the lower whiskers and outliers as β increases.

5.2 Grid environments

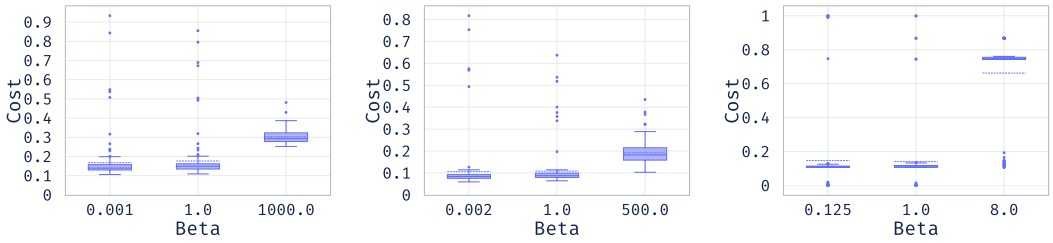
5.2.1 Maximum State Entropy Exploration

The MSEE task integrates a 10×10 grid that comprises two different types of squares: (i) normal terrain; and (ii) difficult terrain. The agent can move in all four directions, ending up in the corresponding adjacent square (if moving out-of-bounds, the agent remains in the same square). When the agent is inside a difficult terrain square, there is a p_{stuck} probability that the agent gets stuck, which translates in canceling the selected (moving) action and thus remaining trapped in the current square. Upon getting stuck, there is an additional p_{unstuck} probability that untraps the agent allowing it to move freely again. As expected, inside normal terrain, the agent's selected moving action is always guaranteed to occur. Due to the environment's stochastic nature, the agent needs to consider if there are benefits in exploring certain difficult terrain squares at the expense of getting stuck, which can have a negative effect on the discounted costs received from that point onward.

In Figure 4a, we report the box plots of the empirical costs obtained by ERM-MCTS with $\beta \in \{0.001, 1, 1000\}$. When $\beta \in \{0.001, 1\}$, ERM-MCTS outputs similar risk-seeking policies, as both display identical Inner Fence Interval (IFI) ranges, inside $[0.1, 0.2]$. Additionally, the expectation value is lower when $\beta = 0.001$, having a value of 0.167, whereas the expectation for $\beta = 1$ is 0.177. Furthermore, both policies exhibit a wide range of outliers on the upper-half of the distribution, implying that in some runs the agent got trapped for long periods of time when covering difficult terrain squares, further alluding to the potential disadvantages of having a risk-seeking behavior. Additionally, the worst outlier is higher for the policy with $\beta = 0.001$, which is expected since this policy is effectively closer to the risk-seeking (or risk-neutral) boundary in the *risk-awareness* spectrum. When comparing the policy obtained when $\beta = 1000$ to the the previous two ($\beta \in \{0.001, 1\}$), the range of the objective distribution becomes more concentrated as there are fewer outliers. This shows the effectiveness of the risk-averse behavior in avoiding the worst outcomes, as the highest cost is only 0.48 when setting $\beta = 1000$, vs. 0.93 using $\beta = 0.001$. The drawback of this policy manifests by inducing an average case that performs worse than policies that are more risk-seeking, where the average of the objective distributions are 0.30 and 0.167, for the policies using $\beta = 1000$ and $\beta = 0.001$, respectively. Finally, we further validate our analysis with Figure 5, which exposes the average number of timesteps spent in each square over all runs. We observe that when $\beta = 1000$, the policy navigates towards the lower-right corner to avoid difficult terrain squares altogether. As β decreases, the heatmap becomes more uniformly distributed showcasing that the policy tries to visit all squares (normal and difficult terrain) more often.

5.2.2 Imitation Learning

We reuse the MSEE environment for our IL task. Additionally, we carefully design a behavior policy that changes from risk-seeking and risk-aware behaviors at well-defined intervals. In high-level terms, the behavior policy starts by exploring a zone containing both types of squares, followed by a period where only normal squares are visited. This procedure repeats a second time to make sure all grid squares are visited once. Furthermore, the behavior policy will not be subjected to the environment's stochasticity, i.e., it will never get trapped when visiting difficult terrain. In this manner, we guarantee that achieving a complete match will be extremely unlikely, further leading the ERM-MCTS policy to consider when and what risks to take.



(a) MSEE (b) IL (c) MO

Figure 4: Grid environments: box plots obtained, for different tasks, by deploying ERM-MCTS with different β values. Box plots computed for 128 independent runs. Lower is better.

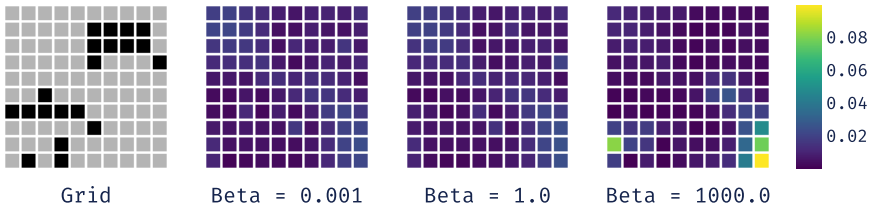


Figure 5: MSEE (grid environment): leftmost grid illustrates the environment, where normal and difficult terrain are depicted in gray and black squares, respectively; the following grids showcase the average time spent in each square averaged over all (128) runs, for each $\beta \in \{0.001, 1, 1000\}$.

To analyze the IL task, we compare ERM-MCTS with three different beta values, $\beta \in \{0.02, 1, 500\}$. As shown in Figure 4b, we clearly observe the same pattern reported in the MSEE task (Sec. 5.2.1): as β increases, the maximum cost obtained decreases while the expectation and median increase. Here, the expectation obtains the values 0.106, 0.109, and 0.192, while the median acquires 0.08, 0.09, and 0.18 for the policies with $\beta = 0.02$, $\beta = 1$, and $\beta = 500$, respectively. Conversely, and for the same policies, the uppermost outliers are 0.81, 0.65, and 0.44, respectively. To get a closer look at how behaviors differ as β changes, we plot, in Figure 6, the average number of runs visiting difficult terrain squares in each timestep. Interestingly, the policy with the highest β emphasizes matching the occupancy of normal squares during the first 40 time steps, as there is a negligible number of runs stepping inside difficult terrain. For the remaining time, the policy focuses more on equalizing the occupancy at difficult terrain squares, as outlined by the increased number of runs exploring such squares. We argue that such behavior is due to discounting, since it is more forgiving to get trapped closer to the end of the episode, considering that it has less impact on the discounted return. Contrastingly, the other policies ($\beta \in \{0.02, 1\}$) start visiting difficult terrain squares from the beginning of the episode. Consequently, the likelihood of getting trapped sooner and for longer periods of time increases, as hinted in Figure 4b by the wide range of outliers on the upper half of the cost distribution.

5.2.3 Multi-Objective

For the MO task, we extend the resource-gathering environment (Barrett & Narayanan, 2008), aiming at introducing challenging probabilistic dynamics that allow the modulation of policies demonstrating behaviors with different degrees of risk-awareness. In the modified environment, the agent's objective consists in retrieving two different resources (R_1 and R_2) back to its starting location while avoiding enemies present at specific squares. Additionally, we set a slippery dynamic, where with probability p_{slip} the agent will move perpendicularly to the selected direction. All agent objectives are modulated using the following reward functions: (i) agent receives a cost when failing to deliver R_1 to the home location (the agent's initial position); (ii) agent receives a cost when failing to delivering R_2 to the home location; (iii) agent receives a cost when getting defeated by an enemy, which

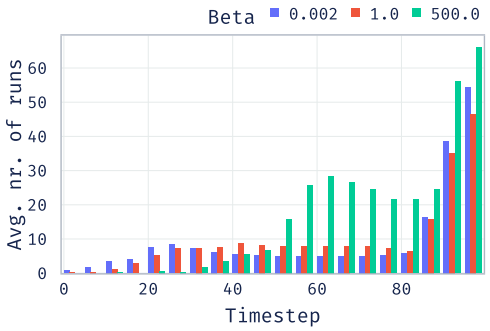


Figure 6: IL (grid environment): Average number of runs visiting difficult terrain squares in each timestep for different β values.

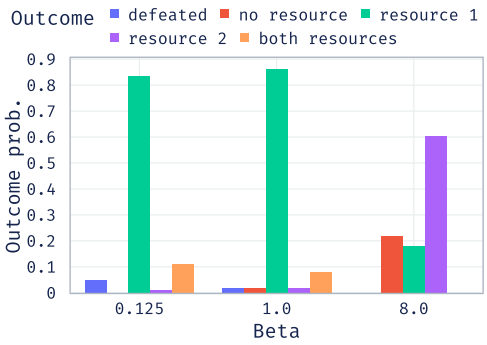


Figure 7: MO (grid environment): Probability of each outcome for different β values.

happens, with probability p_{defeat}, when visiting a square containing an enemy. Finally, we combine the three independent reward functions using a non-linear mapping that prioritizes collecting R_1 over the other ones. We refer to Appendix 11.5.1 for a complete description of the dynamics, cost functions, and hyperparameters used.

We simulate ERM-MCTS with $\beta \in \{0.125, 1, 8\}$. As shown in Figure 4c, the cost distributions obtained by the policies where $\beta \in \{0.125, 1\}$ are highly identical. We again observe the risk-seeking behavior of these policies, since, on average, the costs obtained remain concentrated near the lower bound. For instance, the expectation is 0.16 and 0.18 for $\beta = 0.125$ and $\beta = 1$, respectively. The IFI is $[0.10, 0.14]$ in both cases. Furthermore, since during the optimization of the ERM's objective, the expectation outweighs the upper tail of the distribution, eventually some runs conclude at the worst possible outcome, in which the agent gets defeated by an enemy. Contrastingly, the policy generated for $\beta = 8$ clearly eliminates this possible outcome. In this case, the agent never gets the cost of 1 by inducing a risk-aware behavior that most of the time prefers not to collect R_1 due to its proximity to enemies and, as such, the likelihood of defeat. Therefore, the median centers around 0.87, indicating the agent never retrieved any resources to its initial position. Finally, in Figure 7, we show in more detail the frequency of outcomes that each policy achieves. As hinted previously, when $\beta \in \{0.125, 1\}$ we observe a small fraction of runs where the agent gets defeated. On the other hand, the risk-averse policy ($\beta = 8$) disallows this outcome entirely by, most of the time (82%), either avoiding gathering any resources or just retrieving R_2.

6 Conclusion

We motivate, propose and formalize risk-aware GUMDPs, which allow to take advantage of the flexibility of the GUMDPs framework with respect to objective specification, while trading off expected performance and risk-aversion. To solve risk-aware GUMDPs, we first explore a connection between risk-aware GUMDPs and solving a particular risk-aware MDP, named occupancy MDP, in which the agent keeps track of the empirical frequency of visitation of state-action pairs up to the current timestep. We propose a provably correct online planning approach based on an MCTS algorithm to solve the risk-aware occupancy MDP, effectively solving the original risk-aware GUMDP problem up to any desired accuracy. We provide a set of experimental results under a set of diverse tasks showing that our approach successfully trades off risk-neutral and risk-averse behavior.

Future work could investigate whether a similar approach to ours can be used to solve GUMDPs with CVaR objectives (Rockafellar et al., 2000; Chow et al., 2015). Other interesting direction is to investigate whether our approach can be extended to deal with very large/inherently continuous state spaces, e.g., by borrowing ideas from successor features (Barreto et al., 2017; Borsa et al., 2018).

Acknowledgments

This work was supported by Portuguese national funds through the Portuguese Fundação para a Ciência e a Tecnologia (FCT) under projects UID/50021/2025 and UID/PRR/50021/2025 (INESC-ID multi-annual funding), as well as AI-PackBot (project number 14935, LISBOA2030-FEDER-00854700). Pedro P. Santos acknowledges the FCT PhD grant 2021.04684.BD and Fábio Vital acknowledges the FCT PhD grant 2022.14163.BD. Alberto Sardinha acknowledges the CNPq Research Productivity Fellowship (PQ), with reference 312699/2025-5. The authors thank the lab managers at GAIPS for the support provided when running the computational experiments of this work. The authors also thank Jacopo Silvestrin for discussions on earlier versions of this work and Zita Marinho for feedback on the manuscript.

References

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pp. 1, 2004.

David Abel, Will Dabney, Anna Harutyunyan, Mark K. Ho, Michael L. Littman, Doina Precup, and Satinder Singh. On the expressivity of markov reward, 2022.

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms, 2018.

E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.

Philippe Artzner, Freddy Delbaen, Eber Jean-Marc, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9:203 – 228, 07 1999.

Andre Barreto, Will Dabney, Remi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 41–47, 2008.

Nicole Bäuerle and Jonathan Ott. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research*, 74(3):361–379, Dec 2011. ISSN 1432-5217.

Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023.

Aharon Ben-Tal and Marc Teboulle. Expected utility, penalty functions, and duality in stochastic nonlinear programming. *Management Science*, 32(11):1445–1466, 1986.

V. S. Borkar and S. P. Meyn. Risk-sensitive optimal control for markov decision processes with monotone cost. *Mathematics of Operations Research*, 27(1):192–209, 2002.

Diana Borsa, André Barreto, John Quan, Daniel J. Mankowitz, Rémi Munos, Hado van Hasselt, David Silver, and Tom Schaul. Universal successor features approximators. *CoRR*, abs/1812.07626, 2018.

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Y.S. Chow, H. Robbins, and D. Siegmund. *Great Expectations: The Theory of Optimal Stopping*. 1971. ISBN 9780395053140.

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. The inventory problem: Ii. case of unknown distributions of demand. *Econometrica*, 20(3):450–466, 1952.

Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *CoRR*, abs/2003.02189, 2020.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function, 2018.

H. Föllmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time*. De Gruyter Textbook. 2016. ISBN 9783110463453.

Javier García, Fern, and o Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015.

Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Olivier Bachem, Rémi Munos, and Olivier Pietquin. Concave utility reinforcement learning: the mean-field game viewpoint, 2022.

Conor F. Hayes, Mathieu Reymond, Diederik M. Roijers, Enda Howley, and Patrick Mannion. Monte carlo tree search algorithms for risk-aware and multi-objective reinforcement learning, 2022.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2681–2691, 2019.

Ronald A. Howard and James E. Matheson. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, March 1972.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), apr 2017.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. DOI: 10.1017/9781108571401.

T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016.

Jia Lin Hau, Marek Petrik, and Mohammad Ghavamzadeh. Entropic risk optimization in discounted MDPs. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pp. 47–76. PMLR, 25–27 Apr 2023.

Alexandre Marthe, Samuel Bounan, Aurélien Garivier, and Claire Vernade. Efficient risk-sensitive planning via entropic risk measures, 2025.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Oliver Mortensen and Mohammad Sadegh Talebi. Entropic risk optimization in discounted mdps: Sample complexity bounds with a generative model, 2025.

Mirco Mutti, Riccardo De Santi, Piersilvio De Bartolomeis, and Marcello Restelli. Convex reinforcement learning in finite trials. *Journal of Machine Learning Research*, 24(250):1–42, 2023.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1–2): 1–179, 2018. ISSN 1935-8261.

Bernardo Pagnoncelli, Oscar Dowson, and David Morton. Multistage stochastic programs with the entropic risk measure. February 2022.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Roxana Radulescu, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. Multi-objective multi-agent decision making: A utility-based analysis and survey. *CoRR*, abs/1909.02964, 2019.

R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

{Diederik M.} Roijers, Denis Steckelmacher, and Ann Nowé. Multi-objective reinforcement learning for the expected utility of the return. 2020. 2018 Adaptive Learning Agents, ALA 2018 - Co-located Workshop at the Federated AI Meeting, FAIM 2018.

Pedro P. Santos, Alberto Sardinha, and Francisco S. Melo. The number of trials matters in infinite-horizon general-utility markov decision processes, 2024.

Pedro P. Santos, Alberto Sardinha, and Francisco S. Melo. Solving general-utility markov decision processes in the single-trial regime with online planning, 2025.

Pedro P. Santos, Jacopo Silvestrin, Alberto Sardinha, and Francisco S. Melo. Entropic risk-aware monte carlo tree search, 2026.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Shaler Stidham. Socially and individually optimal control of arrivals to a gi/m/1 queue. *Management Science*, 24(15):1598–1610, 1978.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex mdps. *CoRR*, abs/2106.00661, 2021.

Junyu Zhang, Amrit Singh Bedi, Mengdi Wang, and Alec Koppel. Cautious reinforcement learning via distributional risk in the dual domain, 2020.

Supplementary Materials

The following content was not necessarily subject to peer review.

7 Extended related work discussion

Previous works studied GUMDPs with risk-neutral objectives (Zahavy et al., 2021; Geist et al., 2022). Zahavy et al. (2021) reformulate GUMDPs as a two-player game involving a policy and a cost (negative reward) player, using Fenchel duality. Geist et al. (2022) connect GUMDPs and mean-field games. Still in the risk-neutral setting, subsequent works identified a key implicit assumption underlying the GUMDPs framework: the performance of a given policy may depend on the number of trials/trajectories drawn to evaluate its performance (Mutti et al., 2023; Santos et al., 2024). In fact, the previous works show that the standard formulation of GUMDPs implicitly assumes the performance of a given policy is evaluated under an infinite number of trials/trajectories, an assumption that may be violated under many interesting application domains. To address this gap, Mutti et al. (2023); Santos et al. (2025) introduce finite trials formulations for GUMDPs, allowing to find optimal risk-neutral policies with respect to a finite number of trials/trajectories. None of the works above study risk-aware settings, being focused on learning risk-neutral policies. As pointed out in the main text, some of the results in our paper build on top of results put forth by the aforementioned studies. As an example, our work relies on the occupancy MDP formulation, as described in Sec. 4.2, which was introduced in (Santos et al., 2025).

In the context of risk-aware MDPs, previous works studied the optimization of risk-aware objectives such as the conditional value-at-risk Chow et al. (2015) or the ERM (Lin Hau et al., 2023; Marthe et al., 2025; Mortensen & Talebi, 2025). In the context of our work, we are focused on more general objective functions of the occupancies induced by the different policies, not only being focused on the case of linear objectives (MDPs). Nevertheless, some of the techniques we employ resemble those adopted in the context of risk-aware MDPs. As an example, it is common in the context of risk-aware MDPs to extend the state space to accommodate sufficient statistics since the optimal policy depends on the history only through a certain kind of "sufficient statistic" (Bäuerle & Ott, 2011). This resembles the construction of our occupancy MDP; however, the states in the occupancy MDP do not correspond to sufficient statistics of the history but instead correspond directly to histories (Lemma 4.3). This is required since, under more general settings, optimal policies may depend arbitrarily on the history. Other works, such as Lin Hau et al. (2023), exploit discounting and consider finite-horizon objectives to compute approximately optimal policies for infinite-horizon objectives - this technique is common in the field, and we followed a similar approach in 4.2.

Finally, in the context of MDPs, Zhang et al. (2020) propose a new definition of the risk of a policy (called caution) as a function of the occupancy induced by the policy. Then, the authors formulate a caution-sensitive policy optimization problem by adding the caution risk as a penalty function to the dual objective of the linear programming formulation of MDPs. The formulation proposed by the authors can itself be seen as solving a risk-neutral GUMDP, where the objective function contains a term that captures the risk of a given policy. In our work, we are instead focused on solving risk-aware GUMDPs.

8 Lipschitz constants

Standard MDP ($f(\boldsymbol{d}) = \boldsymbol{c}^\top \boldsymbol{d}$) It holds that

$$|f(\boldsymbol{d}_1) - f(\boldsymbol{d}_2)| = |\boldsymbol{c}^\top(\boldsymbol{d}_1 - \boldsymbol{d}_2)| \leq \sum_{s,a} |c(s,a)||d_1(s,a) - d_2(s,a)| \leq \max_{s,a} |c(s,a)| \|\boldsymbol{d}_1 - \boldsymbol{d}_2\|_1.$$

Maximum state entropy exploration ($f(\boldsymbol{d}) = \boldsymbol{d}^\top \log(\boldsymbol{d})$) We assume $\boldsymbol{d}$ is lower bounded by ϵ, i.e., $d(s,a) \geq \epsilon$ with $0 < \epsilon < e^{-2}$ for all $s \in \mathcal{S}, a \in \mathcal{A}$. We let $f(\boldsymbol{d}) = \sum_{s,a} g(d(s,a))$, for $g(x) =$

Table 1: Lipschitz constants for common objective functions found in the GUMDPs literature. In (†) we assume $\boldsymbol{d}$ is lower bounded by ϵ satisfying $0 < \epsilon < e^{-2}$.

Task	**Objective** ($f(\boldsymbol{d})$)	**Lipschitz constant** (L)
MDPs/RL	$\boldsymbol{d}^\top \boldsymbol{c}, \quad \boldsymbol{c} \in \mathbb{R}^{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}$	$\max_{s,a} \lvert c(s,a)\rvert$
Max. state entropy expl.	$\boldsymbol{d}^\top \log(\boldsymbol{d})$	$\lvert \log(\epsilon) + 1\rvert$ (†)
Imitation learning	$\lVert \boldsymbol{d} - \boldsymbol{d}_{\pi_b} \rVert_2^2, \quad \boldsymbol{d}_{\pi_b} \in \Delta(\mathcal{S} \times \mathcal{A})$	4
Multi-objective MDPs (weighted)	$n_1 \boldsymbol{d}^\top \boldsymbol{c}_1 + \ldots + n_k \boldsymbol{d}^\top \boldsymbol{c}_k$ $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \in \mathbb{R}^{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}$ $n_1, \ldots, n_k \in \mathbb{R}$	$\max_{s,a} \lvert \tilde{c}(s,a)\rvert$ $\tilde{\boldsymbol{c}} = n_1 \boldsymbol{c}_1 + \ldots + n_k \boldsymbol{c}_k$
Multi-objective MDPs (max)	$f(\boldsymbol{d}) = \max_{i \in \{1,\ldots,k\}} \boldsymbol{d}^\top \boldsymbol{c}_i$ $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \in \mathbb{R}^{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}$	$\max_{i \in \{1,\ldots,k\}} L_i$ $L_i = \max_{s,a} \lvert c_i(s,a)\rvert$
Multi-objective MDPs (min)	$f(\boldsymbol{d}) = \min_{i \in \{1,\ldots,k\}} \boldsymbol{d}^\top \boldsymbol{c}_i$ $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \in \mathbb{R}^{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}$	$\max_{i \in \{1,\ldots,k\}} L_i$ $L_i = \max_{s,a} \lvert c_i(s,a)\rvert$

$x \log(x)$. We note that, $g'(x) = \log(x) + 1$ and it holds for any $x \in [\epsilon, 1]$ that $|g'(x)| \leq |\log(\epsilon) + 1|$. Thus, for any $x_1, x_2 \in [\epsilon, 1]$ we have that

$$
\begin{aligned}
|g(x_1) - g(x_2)| &= \left| \int_{x_2}^{x_1} g'(x) dx \right| = \left| \int_{\min\{x_1,x_2\}}^{\max\{x_1,x_2\}} g'(x) dx \right| \\
&\leq \int_{\min\{x_1,x_2\}}^{\max\{x_1,x_2\}} |g'(x)|\, dx \leq \int_{\min\{x_1,x_2\}}^{\max\{x_1,x_2\}} |\log(\epsilon) + 1|\, dx \\
&= |\log(\epsilon) + 1|\, |x_1 - x_2|.
\end{aligned}
$$

Thus, for any $\boldsymbol{d}_1, \boldsymbol{d}_2 \in \Delta(\mathcal{S} \times \mathcal{A})$ lower bounded by $0 < \epsilon < e^{-2}$, it holds that

$$
\begin{aligned}
|f(\boldsymbol{d}_1) - f(\boldsymbol{d}_1)| &= \left| \sum_{s,a} \left(g(d_1(s,a)) - g(d_2(s,a)) \right) \right| \\
&\overset{(a)}{\leq} \sum_{s,a} |g(d_1(s,a)) - g(d_2(s,a))| \\
&\leq \sum_{s,a} |\log(\epsilon) + 1|\, |d_1(s,a) - d_2(s,a)| \\
&= |\log(\epsilon) + 1|\, \|\boldsymbol{d}_1 - \boldsymbol{d}_2\|_1
\end{aligned}
$$

were (a) follows from the triangular inequality.

Imitation learning ($f(\boldsymbol{d}) = \|\boldsymbol{d} - \boldsymbol{d}_{\pi_b}\|_2^2$) It holds that $\nabla f(\boldsymbol{d}) = 2(\boldsymbol{d} - \boldsymbol{d}_{\pi_b})$. Now,

$$
\max_{\boldsymbol{d} \in \Delta(\mathcal{S}\times\mathcal{A})} \|\nabla f(\boldsymbol{d})\|_1 = 2 \max_{\boldsymbol{d} \in \Delta(\mathcal{S}\times\mathcal{A})} \|\boldsymbol{d} - \boldsymbol{d}_{\pi_b}\|_1 \leq 2 \max_{\boldsymbol{d}_1, \boldsymbol{d}_2 \in \Delta(\mathcal{S}\times\mathcal{A})} \|\boldsymbol{d}_1 - \boldsymbol{d}_2\|_1 = 4.
$$

Since the function f is continuous and differentiable over the simplex, which is compact, it holds that $L = 4$ is a valid Lipschitz constant as it corresponds to an upper bound on the maximum magnitude of the gradient of f over $\Delta(\mathcal{S} \times \mathcal{A})$.

Multi-objective MDP, weighted ($f(\boldsymbol{d}) = n_1 \boldsymbol{d}^\top \boldsymbol{c}_1 + \ldots + n_k \boldsymbol{d}^\top \boldsymbol{c}_k$) We note that f can be equivalently rewritten as $f(d) = \boldsymbol{d}^\top \tilde{\boldsymbol{c}}$, where $\tilde{\boldsymbol{c}} = n_1 \boldsymbol{c}_1 + \ldots + n_k \boldsymbol{c}_k$. Therefore

$$|f(\boldsymbol{d}_1) - f(\boldsymbol{d}_2)| \leq \max_{s,a} |\tilde{c}(s,a)| \|\boldsymbol{d}_1 - \boldsymbol{d}_2\|_1.$$

Multi-objective MDP, max ($f(\boldsymbol{d}) = \max_{i \in \{1,\ldots,k\}} \boldsymbol{d}^\top \boldsymbol{c}_i$) The result follows by noting that letting $L = \max_{i \in \{1,\ldots,k\}} L_i$ yields a valid Lipschiz constant, where $L_i = \max_{s,a} |c_i(s,a)|$, since we are taking the maximum of Lipschitz functions with Lipschitz constants $L_1, \ldots, L_k$.

Multi-objective MDP, min ($f(\boldsymbol{d}) = \min_{i \in \{1,\ldots,k\}} \boldsymbol{d}^\top \boldsymbol{c}_i$) It holds that

$$|f(\boldsymbol{d}_1) - f(\boldsymbol{d}_2)| = \left| \min_{i \in \{1,\ldots,k\}} \boldsymbol{c}_i^\top \boldsymbol{d}_1 - \min_{i \in \{1,\ldots,k\}} \boldsymbol{c}_i^\top \boldsymbol{d}_2 \right| \leq \max_{i \in \{1,\ldots,k\}} |\boldsymbol{c}_i^\top \boldsymbol{d}_1 - \boldsymbol{c}_i^\top \boldsymbol{d}_2| \leq L \|\boldsymbol{d}_1 - \boldsymbol{d}_2\|_1,$$

where $L = \max_{i \in \{1,\ldots,k\}} L_i$ and $L_i = \max_{s,a} |c_i(s,a)|$.

9 Supplementary materials for Sec. 4

9.1 Proof of Proposition 4.2

Lemma 9.1. *For any $\omega \in \Omega$, $\pi \in \Pi_{\text{NM}}$ and $H \in \mathbb{N}$ it holds that $|\mathrm{f}^\pi(\omega) - \mathrm{f}^\pi_H(\omega)| \leq 2L_f \gamma^H$.*

Proof. For any $\omega \in \Omega$, $\pi \in \Pi_{\text{NM}}$ and $H \in \mathbb{N}$ it holds that

$$\begin{aligned}
|\mathrm{f}^\pi(\omega) - \mathrm{f}^\pi_H(\omega)| &= \left| f(\mathbf{d}^\pi(\omega)) - f(\mathbf{d}^{\pi,H}(\omega)) \right| \\
&\overset{\text{(a)}}{\leq} L_f \left\| \mathbf{d}^\pi(\omega)) - \mathbf{d}^{\pi,H}(\omega)) \right\|_1 \\
&\overset{\text{(b)}}{=} L_f \left\| (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbf{d}^\pi_t(\omega) - \frac{(1-\gamma)}{1-\gamma^H} \sum_{t=0}^{H-1} \gamma^t \mathbf{d}^\pi_t(\omega) \right\|_1 \\
&= L_f \left\| \frac{(1-\gamma)}{1-\gamma^H} \sum_{t=0}^{H-1} \gamma^t \left((1-\gamma^H) \mathbf{d}^\pi_t(\omega) - \mathbf{d}^\pi_t(\omega) \right) + (1-\gamma) \sum_{t=H}^{\infty} \gamma^t \mathbf{d}^\pi_t(\omega) \right\|_1 \\
&\overset{\text{(c)}}{\leq} L_f \left(\frac{(1-\gamma)}{1-\gamma^H} \sum_{t=0}^{H-1} \gamma^t \left\| (1-\gamma^H) \mathbf{d}^\pi_t(\omega) - \mathbf{d}^\pi_t(\omega) \right\|_1 + (1-\gamma) \sum_{t=H}^{\infty} \gamma^t \|\mathbf{d}^\pi_t(\omega)\|_1 \right) \\
&= L_f \frac{(1-\gamma)}{1-\gamma^H} \gamma^H \sum_{t=0}^{H-1} \gamma^t \left\| \mathbf{d}^\pi_t(\omega) \right\|_1 + L_f \gamma^H \\
&= 2L_f \gamma^H,
\end{aligned}$$

where: (a) is due to the L_f-Lipschitz assumption; in (b) we used $\mathbf{d}^\pi(\omega) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbf{d}^\pi_t(\omega)$ where $\mathrm{d}^\pi_{t,(s,a)}(\omega) = \mathbf{1}(s_t = s, a_t = a)$ denotes the empirical occupancy induced by the trajectory ω at timestep t and $\mathbf{d}^{\pi,H}(\omega) = (1-\gamma)/(1-\gamma^H) \sum_{t=0}^{H-1} \gamma^t \mathbf{d}^\pi_t(\omega)$. Step (c) follows from the triangular inequality. □

Lemma 9.2. *For any policy $\pi \in \Pi_{\text{NM}}$ and $H \in \mathbb{N}$ it holds that $|\rho(\mathrm{f}^\pi_H) - \rho(\mathrm{f}^\pi)| \leq 2L_f \gamma^H$.*

Proof. From Lemma 9.1 we can infer that, for any $\omega \in \Omega$, $\mathrm{f}^\pi(\omega) - 2L_f\gamma^H \leq \mathrm{f}^\pi_H(\omega) \leq \mathrm{f}^\pi(\omega) + 2L_f\gamma^H$. Thus, from the monotonicity of ρ we have that $\rho(\mathrm{f}^\pi - 2L_f\gamma^H) \leq \rho(\mathrm{f}^\pi_H) \leq \rho(\mathrm{f}^\pi + 2L_f\gamma^H)$. Equivalently, $\rho(\mathrm{f}^\pi) - 2L_f\gamma^H \leq \rho(\mathrm{f}^\pi_H) \leq \rho(\mathrm{f}^\pi) + 2L_f\gamma^H$, and the result follows. □

Lemma 9.3. *If $\pi^*_H = \arg\min_{\pi \in \Pi_{\text{NM}}} \rho(\mathrm{f}^\pi_H)$, then it holds that $\mathrm{OptGap}(\pi^*_H) \leq 4L_f \gamma^H$.*

Proof. As shown in Lemma 9.2, $|\rho(\mathrm{f}_H^\pi) - \rho(\mathrm{f}^\pi)| \leq 2L_f\gamma^H$, for arbitrary $\pi \in \Pi_{\text{NM}}$. From such inequality, we can infer that $\rho(\mathrm{f}_H^\pi) - 2L_f\gamma^H \leq \rho(\mathrm{f}^\pi), \forall \pi \in \Pi_{\text{NM}}$, i.e., function $\rho(\mathrm{f}_H^\pi) - 2L_f\gamma^H$ lower bounds function $\rho(\mathrm{f}^\pi)$. Let $\pi_H^* = \arg\min_{\pi\in\Pi_{\text{NM}}} \rho(\mathrm{f}_H^\pi)$. It holds that

$$\rho\left(\mathrm{f}_H^{\pi_H^*}\right) - 2L_f\gamma^H = \min_\pi \rho(\mathrm{f}_H^\pi) - 2L_f\gamma^H \overset{\text{(a)}}{\leq} \min_\pi \rho(\mathrm{f}^\pi) \overset{\text{(b)}}{\leq} \rho\left(\mathrm{f}^{\pi_H^*}\right), \tag{6}$$

where (a) follows from the fact that $\rho(\mathrm{f}_H^\pi) - 2L_f\gamma^H$ lower bounds $\rho(\mathrm{f}^\pi)$; and (b) from the fact that $\min_\pi \rho(\mathrm{f}^\pi) \leq \rho\left(\mathrm{f}^{\pi'}\right), \forall \pi'$ (from the definition of a minimum). Finally, we note that

$$\begin{aligned}
\rho\left(\mathrm{f}^{\pi_H^*}\right) - \left(\rho\left(\mathrm{f}_H^{\pi_H^*}\right) - 2L_f\gamma^H\right) &= \rho\left(\mathrm{f}^{\pi_H^*}\right) - \rho\left(\mathrm{f}_H^{\pi_H^*}\right) + 2L_f\gamma^H \\
&\leq \left|\rho\left(\mathrm{f}^{\pi_H^*}\right) - \rho\left(\mathrm{f}_H^{\pi_H^*}\right)\right| + 2L_f\gamma^H \\
&\leq 4L_f\gamma^H.
\end{aligned}$$

The above and (6) imply that

$$\mathrm{OptGap}(\pi_H^*) = \rho\left(\mathrm{f}^{\pi_H^*}\right) - \min_{\pi'} \rho\left(\mathrm{f}^{\pi'}\right) \leq 4L_f\gamma^H.$$

□

Proof of Proposition 4.2

Proof. Let $\pi_H^* = \arg\min_{\pi\in\Pi_{\text{NM}}} \rho(\mathrm{f}_H^\pi)$, i.e., π_H^* is optimal with respect to the truncated objective. It holds that,

$$\begin{aligned}
\mathrm{OptGap}(\pi) &= \rho(\mathrm{f}^\pi) - \min_{\pi'\in\Pi_{\text{NM}}} \rho\left(\mathrm{f}^{\pi'}\right) \\
&= \left|\rho(\mathrm{f}^\pi) - \min_{\pi'\in\Pi_{\text{NM}}} \rho\left(\mathrm{f}^{\pi'}\right)\right| \\
&\overset{\text{(a)}}{\leq} \left|\rho(\mathrm{f}^\pi) - \rho(\mathrm{f}^{\pi_H^*})\right| + \left|\rho(\mathrm{f}^{\pi_H^*}) - \min_{\pi'\in\Pi_{\text{NM}}} \rho\left(\mathrm{f}^{\pi'}\right)\right| \\
&\overset{\text{(b)}}{\leq} \left|\rho(\mathrm{f}^\pi) - \rho(\mathrm{f}^{\pi_H^*})\right| + 4L_f\gamma^H \\
&\overset{\text{(c)}}{\leq} |\rho(\mathrm{f}^\pi) - \rho(\mathrm{f}_H^\pi)| + \left|\rho(\mathrm{f}_H^\pi) - \rho(\mathrm{f}^{\pi_H^*})\right| + 4L_f\gamma^H \\
&\overset{\text{(d)}}{\leq} 2L_f\gamma^H + \left|\rho(\mathrm{f}_H^\pi) - \rho(\mathrm{f}^{\pi_H^*})\right| + 4L_f\gamma^H \\
&\overset{\text{(e)}}{\leq} 2L_f\gamma^H + \left|\rho(\mathrm{f}_H^\pi) - \rho(\mathrm{f}_H^{\pi_H^*})\right| + \left|\rho(\mathrm{f}_H^{\pi_H^*}) - \rho(\mathrm{f}^{\pi_H^*})\right| + 4L_f\gamma^H \\
&\overset{\text{(f)}}{\leq} 2L_f\gamma^H + \left|\rho(\mathrm{f}_H^\pi) - \rho(\mathrm{f}_H^{\pi_H^*})\right| + 2L_f\gamma^H + 4L_f\gamma^H \\
&= \rho(\mathrm{f}_H^\pi) - \min_{\pi'\in\Pi_{\text{NM}}} \left\{\rho(\mathrm{f}_H^{\pi'})\right\} + 8L_f\gamma^H,
\end{aligned}$$

where (a) follows from adding and subtracting $\rho(\mathrm{f}^{\pi_H^*})$ and applying the triangular inequality; (b) follows from Lemma 9.3; (c) follows from adding and subtracting $\rho(\mathrm{f}_H^\pi)$ and applying the triangular inequality; (d) follows from Lemma 9.2; (e) follows from adding and subtracting $\rho(\mathrm{f}_H^{\pi_H^*})$ and applying the triangular inequality; and (f) follows from Lemma 9.2. □

9.2 Proof of Lemma 4.4

Proof. Random variable f_H^π, as defined in (4), is associated with the probability space $(\Omega, \mathcal{F}, \mathbb{P}_\pi)$ and random variable $\mathrm{J}_{\mathrm{O}}^{\pi_{\mathrm{O}}}$, as introduced in Sec. 4.2, is associated with the probability space

$(\Omega_{\mathrm{O}}, \mathcal{F}_{\mathrm{O}}, \mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}})$. Random variables f^{π}_{H} and $\mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}}$ are equal in distribution if, for any $y \in \mathcal{Y}$,

$$\mathbb{P}_{\pi}\left[\mathrm{f}^{\pi}_{H} \leq y\right] = \mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}}[\mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}} \leq y].$$

We start by noting that, for any trajectory $\omega_{\mathrm{O}} = (\{s_0, \boldsymbol{o}_0\}, a_0, \{s_1, \boldsymbol{o}_1\}, a_1, \ldots, \{s_H, \boldsymbol{o}_H\}) \in \Omega_{\mathrm{O}}$,

$$\begin{aligned}
\mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}}[\omega_{\mathrm{O}}] &= p_{0,\mathrm{O}}(\{s_0, \boldsymbol{o}_0\}) \cdot \pi_{\mathrm{O}}(a_0|\{s_0, \boldsymbol{o}_0\}) \cdot P^{a_0}_{\mathrm{O}}(\{\mathrm{s}_0, \mathbf{o}_0\}, \{\mathrm{s}_1, \mathbf{o}_1\}) \cdot \ldots \\
&\qquad \cdot \pi_{\mathrm{O}}(a_{H-1}|\{s_{H-1}, \boldsymbol{o}_{H-1}\}) \cdot P^{a_{H-1}}_{\mathrm{O}}(\{s_{H-1}, \boldsymbol{o}_{H-1}\}, \{s_H, \boldsymbol{o}_H\}). \\
&\overset{(a)}{=} p_0(s_0) \cdot \mathbf{1}(\boldsymbol{o}_0 = [0, \ldots, 0]) \cdot \pi_{\mathrm{O}}(a_0|\{s_0, \boldsymbol{o}_0\}) \cdot P^{a_0}(s_0, s_1) \cdot \mathbf{1}(\boldsymbol{o}_1 = \sigma(s_0, \boldsymbol{o}_0, a_0)) \cdot \ldots \\
&\qquad \cdot \pi_{\mathrm{O}}(a_{H-1}|\{s_{H-1}, \boldsymbol{o}_{H-1}\}) \cdot P^{a_{H-1}}(s_{H-1}, s_H) \cdot \mathbf{1}(\boldsymbol{o}_H = \sigma(s_{H-1}, \boldsymbol{o}_{H-1}, a_{H-1})) \\
&\overset{(b)}{=} p_0(s_0) \cdot \mathbf{1}(\boldsymbol{o}_0 = [0, \ldots, 0]) \cdot \pi(a_0|h_0) \cdot P^{a_0}(s_0, s_1) \cdot \mathbf{1}(\boldsymbol{o}_1 = \sigma(s_0, \boldsymbol{o}_0, a_0)) \cdot \ldots \\
&\qquad \cdot \pi(a_{H-1}|h_{H-1}) \cdot P^{a_{H-1}}(s_{H-1}, s_H) \cdot \mathbf{1}(\boldsymbol{o}_H = \sigma(s_{H-1}, \boldsymbol{o}_{H-1}, a_{H-1})) \\
&\overset{(c)}{=} \mathbb{P}_{\pi}\left[\omega\right] \cdot P^{a_{H-1}}(s_{H-1}, s_H) \cdot \mathbf{1}(\boldsymbol{o}_0 = [0, \ldots, 0]) \cdot \mathbf{1}(\boldsymbol{o}_1 = \sigma(s_0, \boldsymbol{o}_0, a_0)) \cdot \ldots \\
&\qquad \cdot \mathbf{1}(\boldsymbol{o}_H = \sigma(s_{H-1}, \boldsymbol{o}_{H-1}, a_{H-1})),
\end{aligned}$$

where in (a) we note that component $\boldsymbol{o}$ of the state is initialized as a zero vector and then deterministically evolves according to σ; any sequence of $\boldsymbol{o}$-vectors that does not evolve according to σ has zero probability under probability measure $\mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}}$. In (b) we used the fact that any stationary policy $\pi_{\mathrm{O}} \in \Pi_{\mathrm{S}}$ for $\mathcal{M}_{\mathrm{O}}$ can be mapped to a particular non-Markovian policy $\pi \in \Pi_{\mathrm{NM}}$ in $\mathcal{M}_f$. In (c) we recall that, for $\omega = (s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1})$, $\mathbb{P}_{\pi}\left[\omega\right] = p_0(s_0) \cdot \pi(a_0|h_0) \cdot P^{a_0}(s_0, s_1) \cdot \pi(a_1|h_1) \cdot P^{a_1}(s_1, s_2) \ldots P^{a_{H-2}}(s_{H-2}, s_{H-1}) \cdot \pi(a_{H-1}|h_{H-1})$.

Now, for any $y \in \mathcal{Y}$ and stationary policy $\pi_{\mathrm{O}} \in \Pi_{\mathrm{S}}$, it holds that

$$\begin{aligned}
\mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}}[\mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}} \leq y] &= \mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}}[\{\omega_{\mathrm{O}} : \mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}}(\omega_{\mathrm{O}}) \leq y\}] \\
&= \sum_{\omega_{\mathrm{O}} \in \Omega_{\mathrm{O}}} \mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}}[\omega_{\mathrm{O}}] \mathbf{1}\left(\mathrm{J}^{\pi_{\mathrm{O}}}_{\mathrm{O}}(\omega_{\mathrm{O}}) \leq y\right) \\
&= \sum_{\omega_{\mathrm{O}} \in \Omega_{\mathrm{O}}} \mathbb{P}^{\mathrm{O}}_{\pi_{\mathrm{O}}}[\omega_{\mathrm{O}}] \mathbf{1}\left(f\left(\frac{1-\gamma}{1-\gamma^H}\boldsymbol{o}_H\right) \leq y\right) \\
&= \sum_{\omega_{\mathrm{O}} \in \Omega_{\mathrm{O}}} \mathbb{P}_{\pi}\left[\omega\right] \cdot P^{a_{H-1}}(s_{H-1}, s_H) \cdot \mathbf{1}(\boldsymbol{o}_0 = [0, \ldots, 0]) \cdot \mathbf{1}(\boldsymbol{o}_1 = \sigma(s_0, \boldsymbol{o}_0, a_0)) \cdot \ldots \\
&\qquad \cdot \mathbf{1}(\boldsymbol{o}_H = \sigma(s_{H-1}, \boldsymbol{o}_{H-1}, a_{H-1})) \mathbf{1}\left(f\left(\frac{1-\gamma}{1-\gamma^H}\boldsymbol{o}_H\right) \leq y\right) \\
&\overset{(a)}{=} \sum_{\omega_{\mathrm{O}} \in \Omega_{\mathrm{O}}} \mathbb{P}_{\pi}\left[\omega\right] \cdot P^{a_{H-1}}(s_{H-1}, s_H) \cdot \mathbf{1}(\boldsymbol{o}_0 = [0, \ldots, 0]) \cdot \mathbf{1}(\boldsymbol{o}_1 = \sigma(s_0, \boldsymbol{o}_0, a_0)) \cdot \ldots \\
&\qquad \cdot \mathbf{1}(\boldsymbol{o}_H = \sigma(s_{H-1}, \boldsymbol{o}_{H-1}, a_{H-1})) \mathbf{1}\left(f\left(\mathbf{d}^{\pi,H}(\omega)\right) \leq y\right) \\
&\overset{(b)}{=} \sum_{\omega \in \Omega} \mathbb{P}_{\pi}\left[\omega\right] \mathbf{1}\left(f\left(\mathbf{d}^{\pi,H}(\omega)\right) \leq y\right) \sum_{\boldsymbol{o}_0, \boldsymbol{o}_1, \ldots, \boldsymbol{o}_H \in \mathcal{O}} \sum_{s_H \in \mathcal{S}} P^{a_{H-1}}(s_{H-1}, s_H) \cdot \\
&\qquad \mathbf{1}(\boldsymbol{o}_0 = [0, \ldots, 0]) \cdot \mathbf{1}(\boldsymbol{o}_1 = \sigma(s_0, \boldsymbol{o}_0, a_0)) \cdot \ldots \cdot \mathbf{1}(\boldsymbol{o}_H = \sigma(s_{H-1}, \boldsymbol{o}_{H-1}, a_{H-1})) \\
&\overset{(c)}{=} \sum_{\omega \in \Omega} \mathbb{P}_{\pi}\left[\omega\right] \mathbf{1}\left(f\left(\mathbf{d}^{\pi,H}(\omega)\right) \leq y\right) \\
&= \sum_{\omega \in \Omega} \mathbb{P}_{\pi}\left[\omega\right] \mathbf{1}\left(\mathrm{f}^{\pi}_{H}\left(\omega\right) \leq y\right) \\
&= \sum_{\omega \in \Omega} \mathbb{P}_{\pi}\left[\{\omega : \mathrm{f}^{\pi}_{H}\left(\omega\right) \leq y\}\right] \\
&= \mathbb{P}_{\pi}\left[\mathrm{f}^{\pi}_{H} \leq y\right],
\end{aligned}$$

where in (a) we noted that, for any $\omega_{\mathrm{O}} \in \Omega_{\mathrm{O}}$, $\mathbf{1}\left(f\left(\frac{1-\gamma}{1-\gamma^H}\boldsymbol{o}_H\right) \leq y\right) = \mathbf{1}\left(f\left(\mathbf{d}^{\pi,H}(\omega)\right) \leq y\right)$. In (b), we split the sum over $\omega_{\mathrm{O}} \in \Omega_{\mathrm{O}}$ as a sum over $\omega \in \Omega$, a sum over each possible vector $\boldsymbol{o} \in \mathcal{O}$ across all timesteps, and a sum over the final state $s_H \in \mathcal{S}$ (not included in ω). We also rearranged the sums by noting that some terms do not depend on some of the sums. In (c) we note that the inner sums over the $\boldsymbol{o}$-vectors and s_H equal one. □

9.3 Proof of Theorem 4.5

Proof. From Lemma 4.4, for any horizon $H \in \mathbb{N}$ and policy $\pi \in \Pi_{\mathrm{NM}}$, it holds that $\mathrm{f}_H^{\pi} \overset{D}{=} \mathrm{J}_{\mathrm{O}}^{\pi_{\mathrm{O}}}$, where π_{O} is the stationary policy for $\mathcal{M}_{\mathrm{O}}$ associated with the non-Markovian policy π for $\mathcal{M}_f$. Thus, for any risk-measure $\rho : \mathcal{L} \to \mathbb{R}$, it holds that $\rho(\mathrm{f}_H^{\pi}) = \rho(\mathrm{J}_{\mathrm{O}}^{\pi_{\mathrm{O}}})$. Also, from Lemma 4.3 there exists a one-to-one mapping between non-Markovian policies for $\mathcal{M}_f$ and stationary policies for $\mathcal{M}_{\mathrm{O}}$ since every state in $\mathcal{M}_{\mathrm{O}}$ is uniquely associated with a particular history in $\mathcal{M}_f$ (and vice versa). Given these two results, we have that, for any $\pi \in \Pi_{\mathrm{NM}}$,

$$\mathrm{OptGap}_H(\pi) = \rho\left(\mathrm{f}_H^{\pi}\right) - \min_{\pi' \in \Pi_{\mathrm{NM}}} \rho\left(\mathrm{f}_H^{\pi'}\right) = \rho\left(\mathrm{J}_{\mathrm{O}}^{\pi_{\mathrm{O}}}\right) - \min_{\pi'_{\mathrm{O}} \in \Pi_{\mathrm{S}}} \rho\left(\mathrm{J}_{\mathrm{O}}^{\pi'_{\mathrm{O}}}\right),$$

and the conclusion follows. □

10 ERM-MCTS pseudocode

We display in Algorithm 1 the pseudocode for the ERM-MCTS algorithm.

Algorithm 1 ERM$_\beta$-MCTS for finite-horizon MDPs with terminal costs.

Inputs: s_0 (root node), H (horizon), n (number of MCTS iterations), and $\{\theta_t\}_{t\in\{1,\dots,H-1\}}$ (exploration constants).

Initialization: $\mathcal{X}_{(s_t,a)} = [\,],\ \forall t \in \{0,\dots,H-1\}, s_t \in \mathcal{S}, a \in \mathcal{A}$ (Initialize list to store sampled terminal costs starting from (s_t, a)). $N(s_t) = 0, N(s_t, a) = 0,\ \forall t \in \{0,\dots,H-1\}, s \in \mathcal{S}, a \in \mathcal{A}$ (Initialize visitation counters).

for $j \in n$ **do**
  for $t \in \{0,\dots,H-1\}$ **do**
    if $N(\mathrm{s}_t, a) = 0$ for some $a \in \mathcal{A}$ **then** $\mathrm{a}_t = a$.
    else
      Select an action according to:
      $$\mathrm{a}_t \in \arg\min_{a\in\mathcal{A}} \left\{ \hat{\rho}_{(\mathrm{s}_t,a),N(\mathrm{s}_t,a)} - \theta_t \sqrt{\frac{\sqrt{N(\mathrm{s}_t)}}{N(\mathrm{s}_t,a)}} \right\},$$
      $$\text{where } \hat{\rho}_{(\mathrm{s}_t,a),N(\mathrm{s}_t,a)} = \frac{1}{\beta} \ln\left(\frac{1}{N(\mathrm{s}_t,a)} \sum_{i=1}^{N(\mathrm{s}_t,a)} \exp\left(\beta \mathrm{x}_i^{(\mathrm{s}_t,a)}\right)\right).$$
    end if
    Sample and transition to next state $\mathrm{s}_{t+1} \sim P_{\mathrm{O}}^{\mathrm{a}_t}(\cdot|\mathrm{s}_t)$.
  end for
  At depth H observe terminal cost $c_{\mathrm{O}}(\mathrm{s}_H)$.
  for $t \in \{H-1,\dots,0\}$ **do**
    Update visitation counters:
      $N(\mathrm{s}_t) = N(\mathrm{s}_t) + 1.$
      $N(\mathrm{s}_t, \mathrm{a}_t) = N(\mathrm{s}_t, \mathrm{a}_t) + 1.$
    Store sampled terminal cost:
      Store $\mathrm{x}_{N(\mathrm{s}_t,\mathrm{a}_t)}^{(\mathrm{s}_t,\mathrm{a}_t)} = c_{\mathrm{O}}(\mathrm{s}_H)$ in $\mathcal{X}_{(\mathrm{s}_t,\mathrm{a}_t)}$.
  end for
end for

11 Experimental Results

Our code is available at https://github.com/gh0stwin/risk-aware-gumdp.

11.1 Baselines

In the context of standard MDPs, the ERM-BI baseline exploits the dynamic programming decompositions of the ERM put forth by (Lin Hau et al., 2023). In particular, Lin Hau et al. (2023) show that the optimal value function $V^* = \{V_t^*\}_{t\in\{0,\dots,H\}}$ and the optimal policy $\pi^* = \{\pi_t^*\}_{t\in\{0,\dots,H-1\}}$ satisfy, for all $s \in \mathcal{S}$,

$$V_t^*(s) = \min_{a\in\mathcal{A}} \left\{ \mathrm{ERM}_{\beta\gamma^t}\left(c_t(s,a) + \gamma V_{t+1}^*(\mathrm{s}')\right)\right\},\ \forall t \in \{0,\dots,H-1\},\ V_H^*(s) = c_H(s),$$
$$\pi_t^*(s) = \arg\min_{a\in\mathcal{A}} \left\{ \mathrm{ERM}_{\beta\gamma^t}\left(c_t(s,a) + \gamma V_{t+1}^*(\mathrm{s}')\right)\right\},\ \forall t \in \{0,\dots,H-1\}.$$

In the context of standard MDPs, we perform backward induction from timestep $t = H$ backwards until timestep $t = 0$ to extract the optimal risk-aware policy. We also note that the occupancy MDP is undiscounted (i.e., $\gamma = 1$) and, hence, $\beta\gamma^t = \beta$ for all timesteps.

11.2 Hyperparameters

For implementation purposes, we let the ERM-MCTS parameters $\theta_t = \theta$ for all $t \in \{0, \dots, H-1\}$, see Sec. 4.3. We display in Tables 2 and 3 the hyperparameters used in our experiments, where N denotes the number of independent runs of ERM-MCTS, n is the number of MCTS iterations per timestep for ERM-MCTS, and θ is the ERM-MCTS exploration constant. Under MDPs, we also run N independent runs of the oracle baseline ERM-BI.

Environment	γ **(Discount)**	H **(Horizon)**	N **(Num. runs)**	n **(MCTS iter.)**	θ **(Expl. const.)**
MDP	0.9	20	100	500	1
MSEE	0.9	20	100	500	1
IL	0.9	20	100	2 000	1
MO	0.99	20	100	500	1

Table 2: Hyperparameters used for the illustrative environment experiments.

Environment	γ **(Discount)**	H **(Horizon)**	N **(Num. runs)**	n **(MCTS iter.)**	θ **(Expl. const.)**
MSEE	0.99	200	128	1024	$\sqrt{2}$
IL	0.99	99	128	1024	$\sqrt{2}$
MO	0.99	40	128	1024	$\sqrt{2}$

Table 3: Hyperparameters used for the grid environment experiments.

11.3 Illustrative environments

We display in Fig. 2 an overview of the illustrative environments. In the remainder of this section, we detail the GUMDPs used for each task, regarding their state space, action space, dynamics, and cost function used.

11.3.1 Standard MDP (Fig. 2 (a))

We employ a four-state MDP where the agent needs to tradeoff at the initial state (s_0) between: 1. a risky action (a_0) that can lead to a low-cost state (s_2), but with some probability the agent ends in a high-cost state (s_3); and 2. a safe action (a_1) that deterministically leads to a medium-cost state (s_1). The agent resets to the initial state with 10% probability while not in the initial state. The cost function is given by $\boldsymbol{c} = [c(s_0), c(s_1), c(s_2), c(s_3)]$ where $c(s_0) = 0,\ c(s_1) = \frac{1}{4},\ c(s_2) = \frac{1}{20},\ c(s_3) = 1$. We let $f(\boldsymbol{d}) = \boldsymbol{d}^\top \boldsymbol{c}$.

11.3.2 Maximum State Entropy Exploration (Fig. 2 (b))

The task is similar to the motivating example from Sec. 1, where the agent aims to explore an environment as uniformly as possible but there is chance that the agent transitions to an absorbing state (s_4). The agent starts at state s_1. We let $f(\boldsymbol{d}) = \boldsymbol{d}^\top \log(\boldsymbol{d})$.

11.3.3 Imitation Learning (Fig. 2 (a))

The dynamics of this environment are the same as the standard MDP (Appendix 11.3.1). The agent aims to imitate the empirical occupancy induced by the trajectory of an agent that selected twice the risky action (a_0) under the standard MDP and then selected the safe action (a_1) for all the remainder timesteps. The trajectory to imitate had a rather *lucky* outcome, as it never ended up in

Table 4: Grid environment parameters used in the MSEE and IL task. Grid positions are defined using 0-based indexing and $(0, 0)$ position it at the top-left corner.

Parameter	Value
Grid size	10×10
Agent starting position	$(9, 0)$
Difficult terrain positions	$(6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (8, 3), (9, 3),$ $(1, 5), (1, 6), (1, 7), (1, 8), (2, 5), (2, 6), (2, 7),$ $(2, 8), (7, 5), (3, 5), (3, 9), (9, 1), (5, 2)$
p_{trap}	0.1
p_{untrap}	0.01

the absorbing state (s_3). Hence, the agent needs to trade off between imitating the behavior policy in states (s_0, s_1, s_2) and risking being absorbed into s_3, or only imitating the behavior policy in the *less risky* states (s_0, s_1). To be precise, the empirical occupancy to imitate is $d_{\pi_b}(s_0, a_0) = 0.20605099$, $d_{\pi_b}(s_0, a_1) = 0.30175732$, $d_{\pi_b}(s_1, a_0) = 0.17054104$, $d_{\pi_b}(s_1, a_1) = 0.15004508$, $d_{\pi_b}(s_2, a_0) = 0.10245629$, $d_{\pi_b}(s_2, a_1) = 0.0829896$, $d_{\pi_b}(s_3, a_0) = 0.0$, $d_{\pi_b}(s_3, a_1) = 0.0$. We let $f(\boldsymbol{d}) = \|\boldsymbol{d} - \boldsymbol{d}_{\pi_b}\|_2^2$.

11.3.4 Multi-Objective MDP (Fig. 2 (c))

The environment is inspired by the FishWood environment (Roijers et al., 2020), where the agent needs to trade off between two cost functions. Each cost function penalizes different behaviors. The cost function $\boldsymbol{c}_1$ is defined as $c_1(s_0) = 0$, $c_1(s_1) = -1$, $c_1(s_2) = 0.5$, $c_1(s_3) = 0$, $c_1(s_4) = 0$. The cost function $\boldsymbol{c}_2$ is defined as $c_2(s_0) = 0$, $c_2(s_1) = 0$, $c_2(s_2) = 0$, $c_2(s_3) = -0.2$, $c_2(s_4) = 0.2$. The agent starts at state s_0. Any action at states s_1, s_2, s_3, and s_4 takes the agent deterministically back to the initial state. We consider three utility functions: (i) *weighted*, where $f(\boldsymbol{d}) = \boldsymbol{d}^\top \boldsymbol{c}_1 + \boldsymbol{d}^\top \boldsymbol{c}_2$; (ii) *max*, where $f(\boldsymbol{d}) = \max(\boldsymbol{d}^\top \boldsymbol{c}_1, \boldsymbol{d}^\top \boldsymbol{c}_2)$; and (iii) *min*, where $f(\boldsymbol{d}) = \min(\boldsymbol{d}^\top \boldsymbol{c}_1, 2 \cdot \boldsymbol{d}^\top \boldsymbol{c}_2)$.

11.4 Grid Environments

11.4.1 Maximum State Entropy Exploration

Table 4 contains the hyperparameters used in the environment employed in the MSEE task. The environment dynamics are described in Sec. 5.2.1.

11.5 Imitation Learning

We reuse the environment and corresponding hyperparameters defined in the MSEE task (Appendix 11.4.1). Additionally, we compute the occupancy of the behavioral policy $\boldsymbol{d}_{\pi_b}$ after the agent performing the sequence of actions depicted in Figure 8. As already mentioned in Sec. 5.2.2, we ensure that the behavior policy never gets trapped inside difficult terrain squares. Under this approach we make sure that ERM-MCTS can model different behaviors (with distinct *risk awareness*) by considering which difficult terrain squares to visit, if any.

11.5.1 Multi-Objective

As stated in Sec. 5.2.3, we use the resource-gathering environment (Barrett & Narayanan, 2008) as our multi-objective task. Here, the agent has three objectives to maximize. The first two objectives address picking and delivering resources R_1 and R_2 to the home location, respectively. The last objective attends to the possibility of the agent getting defeated when stepping inside squares that contain enemies. Aiming to model the necessary information that is required for the reward functions, we define the environment state as a 5-tuple, $s = ((x, y), h_{R_1}, h_{R_2}, h_{\text{start}}, h_{\text{enemy}})$. Where

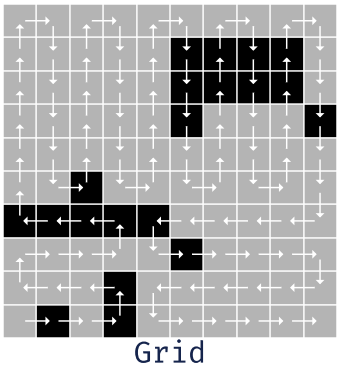


Figure 8: IL (grid environment): Sequence of actions taken by the behavior policy π_b. Agent starts in position $(9, 0)$, assuming 0-based indexing and $(0, 0)$ position it at the top-left corner.

$(x, y) \in \mathbb{N}^2$ holds the agent's position, $h_{R_1}, h_{R_2} \in \{0, 1\}$ assign if the agent is carrying the corresponding resource, $h_{\text{start}} \in \{0, 1\}$ indicates whether the agent left the initial position, and $h_{\text{enemy}} \in \{0, 1\}$ indicates if a *clash* with an enemy occurred. Regarding environment dynamics, the agent can move in all directions, plus we define a slippery condition where, with p_{slip} probability, the agent moves instead perpendicularly to the selected direction. Additionally, resources are collected when the agent enters the resource's square, setting $h_r = 1$, where $r \in \{R_1, R_2\}$. Furthermore, when the agent steps inside a square that contains an enemy, with p_{defeat} probability, h_{enemy} becomes 1. In the case when $h_{\text{enemy}} = 1$, the agent loses immediately and transitions to an absorbing state until the end of the episode. Regarding reward modeling, we define three different reward functions $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_3 \in \mathbb{R}^{|S||A|}$, where all entries are 0 except in specific states:

(i) For $\boldsymbol{r}_1$, the states $s \in \{((x_{\text{home}}, y_{\text{home}}), 1, 0, 1, 0), ((x_{\text{home}}, y_{\text{home}}), 1, 1, 1, 0)\}$ receive 1 to compensate for delivering R_1 to the agent's home position $(x_{\text{home}}, y_{\text{home}})$.

(ii) Analogously, $\boldsymbol{r}_2$ accounts for the cases where R_2 gets transported to $(x_{\text{home}}, y_{\text{home}})$, thus assigning the states $s \in \{((x_{\text{home}}, y_{\text{home}}), 0, 1, 1, 0), ((x_{\text{home}}, y_{\text{home}}), 1, 1, 1, 0)\}$ with reward 1.

(iii) To account for when the agent loses as the consequence of an enemy clash, $\boldsymbol{r}_3$ sets the reward to -1 for any state s having $h_{\text{enemy}} = 1$.

Subsequently, we combine all rewards by applying the following non-linear function, $f(\boldsymbol{d}) = \left(\boldsymbol{r}_1^\top \boldsymbol{d}\right)^{\frac{1}{2}} + \boldsymbol{r}_2^\top \boldsymbol{d} + \boldsymbol{r}_3^\top \boldsymbol{d}$. With $f(\boldsymbol{d})$[1], we assign higher priority when retrieving R_1, e.g., simulating it being more valuable than R_2.

The environment hyper-parameters are defined in Table 5.

11.6 Compute

The illustrative environments were simulated on a laptop CPU (Intel 11th Gen i5-1135G7) with 16 GB of RAM. Additionally, the experiments for the grid environments were deployed on a server with a dual CPU (AMD EPYC 9224 24-Core) and 770 GB of RAM. In the latter, due to the large number of CPUs, we effectively parallelize every experiment, allocating one CPU per independent run. Tables 6 and 7 display the runtime obtained when running ERM-MCTS on the illustrative and grid environments, respectively.

[1]Transforming this objective to handle costs, we first set $\boldsymbol{c}_i = -\boldsymbol{r}_i, \forall i \in \{1, \ldots, 3\}$ and then use $f(\boldsymbol{d}) = \text{sign}(\boldsymbol{c}_1^\top \boldsymbol{d})|\boldsymbol{c}_1^\top \boldsymbol{d}|^{\frac{1}{2}} + \boldsymbol{c}_2^\top \boldsymbol{d} + \boldsymbol{c}_3^\top \boldsymbol{d}$.

Table 5: Grid environment parameters for the resource-gathering environment (MO task). Grid positions are defined using 0-based indexing and $(0, 0)$ position it at the top-left corner.

Parameter	Value
Grid size	5×5
Agent starting position	$(4, 2)$
Home position $(x_{\text{home}}, y_{\text{home}})$	$(4, 2)$
R_1 position	$(0, 2)$
R_2 position	$(1, 4)$
Enemies positions	$(0, 3), (1, 2)$
p_{slip}	0.05
p_{defeat}	0.025

Table 6: ERM-MCTS runtime on illustrative environments.

Environment	Runtime
MDP	(140.50 ± 6.22) s
MSEE	(165.75 ± 25.73) s
IL	(897.75 ± 66.13) s
MO (weighted)	(136.50 ± 1.80) s
MO (max)	(134.75 ± 7.76) s
MO (min)	(142.25 ± 4.76) s

Table 7: ERM-MCTS runtime on grid environments.

Environment	Runtime
MSEE	(2302.19 ± 30.89) s
IL	(593.00 ± 10.35) s
MO	(318.18 ± 17.31) s