Safe Policy Improvement for POMDPs via Finite-State Controllers

Thiago D. Simão*, **Marnix Suilen***, **Nils Jansen**

Department of Software Science
Radboud University
Nijmegen, The Netherlands
{thiago.simao, marnix.suilen, nils.jansen}@ru.nl

Abstract

We study *safe policy improvement* (SPI) for partially observable Markov decision processes (POMDPs). SPI is an offline reinforcement learning (RL) problem that assumes access to (1) historical data about an environment, and (2) the so-called *behavior policy* that previously generated this data by interacting with the environment. SPI methods neither require access to a model nor the environment itself, and aim to reliably improve upon the behavior policy in an offline manner. Existing methods make the strong assumption that the environment is fully observable. In our novel approach to the SPI problem for POMDPs, we assume that a finite-state controller (FSC) represents the behavior policy and that finite memory is sufficient to derive optimal policies. This assumption allows us to map the POMDP to a finite-state fully observable MDP, the *history MDP*. We estimate this MDP by combining the historical data and the memory of the FSC, and compute an improved policy using an off-the-shelf SPI algorithm. The underlying SPI method constrains the policy space according to the available data, such that the newly computed policy only differs from the behavior policy when sufficient data is available. We show that this new policy, converted into a new FSC for the (unknown) POMDP, outperforms the behavior policy with high probability. Experimental results on several well-established benchmarks show the applicability of the approach, even in cases where finite memory is not sufficient.

1 Introduction

Reinforcement learning (RL) is a standard approach to solve sequential decision-making problems when the environment dynamics are unknown (Sutton and Barto 1998). Typically, an RL agent interacts with the environment and optimizes its behavior according to the environment's feedback. However, in offline RL (Levine et al. 2020), the RL agent receives a fixed dataset of past interactions between a behavior policy and the environment and derives a new policy without direct interactions with the environment. One of the challenges in offline RL is to ensure that the new policy outperforms the behavior policy (Cheng et al. 2022). This problem is called *safe policy improvement* (SPI; Thomas, Theocharous, and Ghavamzadeh 2015). Most of the approaches to SPI assume fully observable environments, *e.g.*, (Petrik, Ghavamzadeh, and Chow 2016; Laroche, Trichelair, and des Combes 2019).

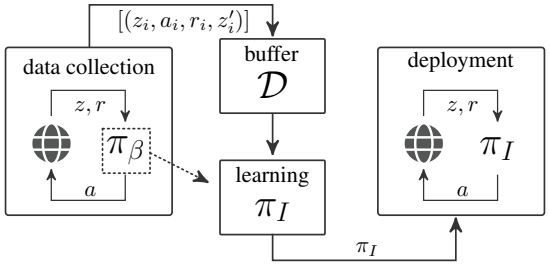

Figure 1: Illustration of the offline reinforcement learning problem in partially observable environments (adapted from Levine et al. 2020). The dashed arrow indicates the setting where the behavior policy is available during learning.

The restriction to full observability poses a serious limitation on the applicability of SPI, as most real-world problems are *partially observable*, due to, for instance, noisy sensors (Kochenderfer 2015). *Partially observable Markov decision processes* (POMDPs) are the standard model for decision-making problems under partial observability (Kaelbling, Littman, and Cassandra 1998). So far, SPI for POMDPs was only studied for memoryless policies (Thomas, Theocharous, and Ghavamzadeh 2015; Yeager et al. 2022). However, POMDP policies often require a notion of memory. In general, optimal policies for POMDPs with infinite horizons require infinite memory, rendering this problem undecidable (Madani, Hanks, and Condon 2003). Nevertheless, finite memory can make good approximations of the optimal policy (Bonet 2002) and are often used in practice for being more explainable (Dujardin, Dietterich, and Chadès 2017). Policies with finite memory may take the form of *finite-state controllers* (FSCs; Meuleau et al. 1999a; Junges et al. 2018; Carr, Jansen, and Topcu 2021).

Our approach. We contribute a novel SPI approach for POMDPs. First, to account for the inherent memory requirement in partially observable domains, we consider a behavior policy represented by a FSC. To create a tractable method, we assume that there exists a finite-memory policy for the POMDP that is optimal, also known as the finite-history-window approach (Kaelbling, Littman, and

*These authors contributed equally.

Moore 1996, Section 7.3). This assumption allows us to cast the POMDP as an equivalent, fully observable, *history MDP* that is finite, instead of the standard infinite-history MDP (Silver and Veness 2010). We are then able to reliably estimate the transition and reward models of this finite-history MDP from the available data. We employ a specific SPI method for MDPs, called safe policy improvement with baseline bootstrapping (SPIBB; Laroche, Trichelair, and des Combes 2019). In particular, we compute an improved policy that outperforms the behavior policy up to an *admissible performance loss* with high probability. In comparison to the approach for mere MDPs (Laroche, Trichelair, and des Combes 2019), we derive an improved bound on this admissible performance loss by exploiting the specific structure of the history MDP. Figure 1 illustrates our approach.

Real-world applications. This setting captures multiple applications, such as predictive maintenance (Andriotis and Papakonstantinou 2021), conservation of endangered species (Chadès et al. 2012), and management of invasive species (Chadès et al. 2011). We may, for instance, have data from the degradation process of a certain asset, which includes logs of inspections and maintenance that were performed according to a fixed schedule (represented, for instance, as a finite-state controller). Once we acquire a new asset, we can formalize the optimization problem with offline RL to compute a new schedule, using the original schedule as a behavior policy.

We demonstrate the applicability of our method on three standard POMDP problems. The evaluation confirms the theoretical findings of our SPIBB approach, in comparison to standard offline RL. We highlight results for varying sizes of memory, and show that we can achieve reliable performance improvement even for problems where finite memory is not sufficient in general.

2 Background

For a countable set X we write $|X|$ for the number of elements in X, and $\Delta(X)$ for the set of probability distributions over X with finite support. Given two probability distributions $P, Q \in \Delta(X)$, the *L1-distance* between P and Q is

$$\|P - Q\|_1 = \sum_{x \in X} |P(x) - Q(x)|.$$

The *L1-error* of an estimated probability distribution $\tilde{P}$ is given by the L1-distance between $\tilde{P}$ and the true distribution P: $\|\tilde{P} - P\|_1$. Finally, we write $\mathbb{I}(x = y)$ for the indicator function returning 1 when $x = y$ and 0 otherwise, and $[l : m]$ for the set of natural numbers $\{l, \dots, m\} \subset \mathbb{N}$.

2.1 MDPs, POMDPs, and FSCs

Definition 1 (POMDP)**.** *A partially observable Markov decision process (POMDP) is a tuple $\mathcal{M} = \langle S, A, T, R, \gamma, Z, O \rangle$, where S, A and Z are finite sets of states, actions, and observations, $T \colon S \times A \to \Delta(S)$ is the transition function, $R \colon S \times A \to [R_{\min}, R_{\max}] \subset \mathbb{R}$ is the reward function with known bounds, $\gamma \in [0, 1) \subset \mathbb{R}$ is the discount factor, and $O \colon S \times A \to \Delta(Z)$ is the observation function.*

As a special case, we have the (fully observable) Markov decision process (MDP; Puterman 1994), where $Z = S$ and $O(z \mid s, a) = \mathbb{I}(z = s)$, so it can be defined as a POMDP without observations: $M = \langle S, A, T, R, \gamma \rangle$.

A *history* is a sequence of observations and actions: $h \in (Z \times A)^* \times Z$. We denote the set of all histories by $\mathcal{H}$, and $\mathcal{H}_k$ denotes all histories of maximal length k, where the length $|h|$ is the number of observations in the history h.

A *belief* $b \in \Delta(S)$ is a distribution over the states of a POMDP. Beliefs are *sufficient statistics* for histories in POMDPs (Åström 1965; Smallwood and Sondik 1973). That is, they provide just as much information as the histories themselves. A belief b can be updated into a new belief b' upon taking an action a and receiving an observation z by performing a Bayesian *belief update* (Kaelbling, Littman, and Cassandra 1998). The belief b' of being in state s given some history h', denoted $b(s \mid h')$, can be recursively computed by repeated applications of the belief update on actions a and observations z in the history $h' = haz$:

$$b'(s' \mid haz) = b'(s' \mid b(\cdot \mid h), a, z).$$

until the history h is empty, denoted $\emptyset$, and where $b(\cdot \mid \emptyset)$ is the initial belief.

A POMDP is equivalent to an infinite-state fully observable MDP called the *history MDP* (Silver and Veness 2010).

Definition 2 (History MDP)**.** *The fully observable* history MDP *of a POMDP is $\langle \mathcal{H}, A, T_{\mathcal{H}}, R_{\mathcal{H}}, \gamma \rangle$, where $\mathcal{H}$ is the set of all histories, A and γ are the actions and discount factor from the POMDP, and $T_{\mathcal{H}} \colon \mathcal{H} \times A \to \Delta(\mathcal{H})$ and $R_{\mathcal{H}} \colon \mathcal{H} \times A \to \mathbb{R}$ are the transition and reward functions:*

$$T_{\mathcal{H}}(haz \mid h, a) = \sum_{s \in S} b(s \mid h) \sum_{s' \in S} T(s' \mid s, a) O(z \mid s', a),$$

$$R_{\mathcal{H}}(h, a) = \sum_{s \in S} b(s \mid h) R(s, a).$$

Since belief states are sufficient statistics for histories, the so-called *belief MDP* of a POMDP serves as a common alternative for the history MDP, we refer to Kaelbling, Littman, and Cassandra (1998) for more details.

A *policy* for a POMDP is a function $\pi \colon \mathcal{H} \to \Delta(A)$, mapping histories to distributions over actions. The set of all policies is Π. The goal is to compute a policy that maximizes the expected discounted reward in the infinite horizon:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r_t\right],$$

where r_t is the reward the agent receives at time step t when following policy π. In general, a policy that maximizes the expected discounted reward requires infinite memory, that is, it needs to account for all possible histories. As such, computing optimal policies in POMDPs is undecidable in general (Madani, Hanks, and Condon 2003).

We may instead use policies with a finite amount of memory. In general, such policies are not optimal, that is, they do not maximize the expected discounted reward, yet they are computationally more tractable. A *finite-memory* policy maps finite histories to actions, $\pi \colon \mathcal{H}_k \to \Delta(A)$, and

can be represented by a finite-state controller of size $\kappa = (|Z|+1)^k$ (Meuleau et al. 1999a; Junges et al. 2018), as we need to account for all possible combinations of observations of size k, with a possible empty observation.

Definition 3 (Finite-state controller)**.** *A* finite-state controller *(FSC) is a tuple* $\langle \mathcal{N}, n^0, \psi, \eta \rangle$ *where* $\mathcal{N}$ *is a finite set of* memory nodes*,* $n^0 \in \mathcal{N}$ *is an* initial node*,* $\psi \colon \mathcal{N} \times Z \to \Delta(A)$ *is an* action mapping*, and* $\eta \colon \mathcal{N} \times Z \times A \to \mathcal{N}$ *is a* memory update function*. A* κ*-FSC is an FSC with* $|\mathcal{N}| = \kappa$*.*

In any time step t, given the current memory node n_t and observation z_t, the action a_t is randomly drawn from the distribution $\psi(\cdot \mid n_t, z_t)$, and the memory node is updated to $n_{t+1} = \eta(n_t, z_t, a_t)$. A policy represented by an FSC can get arbitrarily close to the optimal policy as κ grows (Bonet 2002). Computing finite-state controllers that aim to maximize the expected (discounted) reward can be done in several ways, such as gradient descent (Meuleau et al. 1999b) and convex optimization (Amato, Bernstein, and Zilberstein 2010; Junges et al. 2018; Cubuktepe et al. 2021).

We denote the set of finite-memory policies of size k by Π_k, and the set of finite-memory policies of size k represented by FSCs with some fixed memory update η by Π_k^η. Furthermore, policies of size k can be represented by policies with more memory. These sets are related by the following inclusions, where $k' > k$: $\Pi_k^\eta \subset \Pi_k \subset \Pi_{k'} \subset \Pi$.

Finally, we define the state-based and state-action-based value functions on an (PO)MDP M with policy π as $V_\pi^M(s)$ and $Q_\pi^M(s,a)$ respectively. We omit M or π when they are clear from the context. The *performance* of a policy π in M is denoted by $\rho(\pi, M)$ and is defined as the expected value in some initial state s_0, that is, $\rho(\pi, M) = V_\pi^M(s_0)$. Furthermore, we write $V_{\max}$ for a known upper bound on the absolute value of the performance: $V_{\max} \leq {}^{R_{\max}}/_{1-\gamma}$.

2.2 Safe Policy Improvement on MDPs

Here, we review safe policy improvement (SPI) for MDPs. A *dataset* is a sequence $\mathcal{D}$ of trajectories collected under a *behavior policy* π_β in an MDP M^*. For MDPs, the datasets we consider for SPI are of the form $\mathcal{D} = \langle s_t, a_t, r_t \rangle_{t \in [1:m]}$, and we write $\#_\mathcal{D}(x)$ for the number of times x occurs in $\mathcal{D}$. The goal of SPI is to compute a new policy π_I based on $\mathcal{D}$ that outperforms π_β with an allowed performance loss $\zeta \in \mathbb{R}$ with high probability $1-\delta$.

SPI operates on a set of *admissible MDPs* Ξ, that is, MDPs $M = (S, A, T, R, \gamma)$ which are 'close' to an MDP $\tilde{M} = (S, A, \tilde{T}, \tilde{R}, \gamma)$ estimated from a dataset $\mathcal{D}$ by maximum likelihood estimation (MLE).

Definition 4 (MLE-MDP)**.** *The MLE-MDP of an unknown true MDP* $M^* = (S, A, T, R, \gamma)$ *and a dataset* $\mathcal{D}$ *is a tuple* $\tilde{M} = (S, A, \tilde{T}, \tilde{R}, \gamma)$ *with transition and reward functions* $\tilde{T}$ *and* $\tilde{R}$ *derived from* $\mathcal{D}$ *via maximum likelihood estimation:*

$$\tilde{T}(s' \mid s, a) = \frac{\#_\mathcal{D}(s,a,s')}{\#_\mathcal{D}(s,a)}, \text{ and } \tilde{R}(s,a) = \frac{R_{total}(s,a)}{\#_\mathcal{D}(s,a)},$$

where $R_{total}(s,a) = \sum_{(s_t,a_t,r_t) \in \mathcal{D}} \mathbb{I}(s_t = s \wedge a_t = a) \cdot r_t$ *is the sum of all rewards in the state-action pair* (s,a)*.*

For an MLE-MDP $\tilde{M}$ and error function $e \colon S \times A \to \mathbb{R}$, we define the set of admissible MDPs $\Xi_e^{\tilde{M}}$ with a transition function T that has L_1 distance to the estimated transition function $\tilde{T}$ bounded by the error function e:

$$\Xi_e^{\tilde{M}} = \Big\{ M \mid \forall (s,a). \|T(\cdot \mid s,a) - \tilde{T}(\cdot \mid s,a)\|_1 \leq e(s,a) \Big\}.$$

The general idea behind SPI methods is to define this error function e such that $\Xi_e^{\tilde{M}}$ includes the true MDP M^* with high probability $1-\delta$ (Petrik, Ghavamzadeh, and Chow 2016, Proposition 9). Then one can compute a new policy which is an improvement *for all* MDPs within $\Xi_e^{\tilde{M}}$. An alternative is to simply solve the MLE-MDP, but this could lead to arbitrarily poor policies when the amount of data is insufficient. If, however, the amount of data is sufficient for all state-action pairs, then we can guarantee with high probability that the improved policy computed on the MLE-MDP has a higher performance. Specifically, as pointed out by Laroche, Trichelair, and des Combes (2019), the amount of data is sufficient when for all state-action pairs

$$\#_\mathcal{D}(s,a) \geq \frac{8 V_{\max}^2}{\zeta^2 (1-\gamma)^2} \log \frac{2|S||A|2^{|S|}}{\delta}. \tag{1}$$

Then, with probability $1-\delta$, an optimal policy π_I for $\tilde{M}$ is ζ-approximately safe with respect to the true MDP M^* for some *admissible performance loss* $\zeta \in \mathbb{R}$. That is,

$$\rho(\pi_I, M^*) \geq \rho(\pi^*, M^*) - \zeta \geq \rho(\pi_\beta, M^*) - \zeta,$$

where π^* is an optimal policy in the true MDP M^*. Intuitively, this ensures that the estimated transition function is close enough to the true MDP to guarantee that the policy computed in the MLE-MDP approximately outperforms the behavior policy in the underlying MDP.

2.3 SPI with Baseline Bootstrapping on MDPs

The bound in Equation (1) needs to hold for every state-action pair, which impairs the practical use of optimizing the MLE-MDP. The *SPI with baseline bootstrapping* (SPIBB; Laroche, Trichelair, and des Combes 2019) algorithm overcomes this limitation, allowing the constraint in (1) to be violated on some state-action pairs. These state-action pairs are collected in $\mathcal{U}$, the set of *unknown* state-action pairs with counts smaller than a given hyperparameter $N_\wedge$:

$$\mathcal{U} = \Big\{ (s,a) \in S \times A \mid \#_\mathcal{D}(s,a) \leq N_\wedge \Big\}.$$

SPIBB computes an improved policy π_I on $\tilde{M}$ as above, except that π_I is constrained to follow π_β for unknown state-action pairs: $\forall (s,a) \in \mathcal{U}. \, \pi_I(a \mid s) = \pi_\beta(a \mid s)$. Then, π_I is a ζ-approximately safe improvement of π_β with high probability $1-\delta$:

$$\zeta = \frac{4 V_{\max}}{1-\gamma} \sqrt{\frac{2}{N_\wedge} \log \frac{2|S||A|2^{|S|}}{\delta}} - \rho(\pi_I, \tilde{M}) + \rho(\pi_\beta, \tilde{M}), \tag{2}$$

where ζ is computed via (Theorem 2; Laroche, Trichelair, and des Combes 2019).

3 SPIBB for POMDPs

Now we detail our approach to apply SPIBB to POMDPs.

Formal problem statement. Given a POMDP $\mathcal{M} = \langle S, A, T, R, \gamma, Z, O \rangle$ of which the transition and observation functions are unknown, some initial belief $b \in \Delta(S)$, and a finite-memory behavior policy represented as a κ-FSC $\pi_\beta = (\mathcal{N}, n^0, \psi, \eta)$, the goal is to apply SPIBB to construct a new κ-FSC $\pi_I = (\mathcal{N}, n^0, \psi', \eta)$ with the same nodes and memory structure η, *i.e.* $\pi_\beta, \pi_I \in \Pi_k^\eta$, such that with high probability $1-\delta$, π_I is a ζ-approximately safe improvement over π_β with respect to $\mathcal{M}$. That is, with a probability of at least $1-\delta$ we have

$$\rho(\pi_I, \mathcal{M}) \geq \rho(\pi_\beta, \mathcal{M}) - \zeta.$$

3.1 From POMDP to Finite-History MDP

While a POMDP can be mapped to a fully observable history MDP (Definition 2), this MDP has infinitely many states, making a direct application of SPI(BB) methods infeasible. To mitigate this issue, we make an assumption on the structure of the history MDP (and inherently on the POMDP) that implies that the history MDP is equivalent to a smaller, finite, MDP. We formalize this assumption via stochastic bisimulation (Givan, Dean, and Greig 2003). Intuitively, this bisimulation is an equivalence relation that relates (history) states that *behave* similarly according to reward signals.

Definition 5 (Bisimilarity of history states)**.** *A stochastic bisimulation relation $E \subseteq \mathcal{H} \times \mathcal{H}$ on history states $h_1, h_2 \in \mathcal{H}$ is an equivalence relation satisfying*

$$E(h_1, h_2) \iff \forall a \in A. \quad R_{\mathcal{H}}(h_1, a) = R_{\mathcal{H}}(h_2, a) \text{ and}$$
$$\forall h_1', h_2' \in \mathcal{H} \text{ with } E(h_1', h_2') \text{ we have}$$
$$T_{\mathcal{H}}(h_1' \mid h_1, a) = T_{\mathcal{H}}(h_2' \mid h_2, a).$$

The largest stochastic bisimulation relation is called (stochastic) bisimulation, denoted by $\sim$. We write $[h]_\sim$ for the equivalence class of history h under $\sim$, and $\mathcal{H}/_\sim$ for the set of equivalence classes.

Assumption 1 (Sufficiency of finite histories)**.** *Every history state h of size $|h| > k$ in the history MDP is bisimilar to a history state h' of size $|h'| \leq k$. That is, $h \sim h'$.*

As a consequence, the history MDP satisfying Assumption 1 has a finite bisimulation quotient MDP (Givan, Dean, and Greig 2003), and we call it a *finite-history MDP* instead. This finite-history MDP consists of states that are the equivalence classes of histories under $\sim$. Note that belief remains a sufficient statistic in this case, *i.e.*, $b(s \mid [h]_\sim) = b(s \mid h)$.

Definition 6 (Finite-history MDP)**.** *A POMDP satisfying Assumption 1 is a fully observable finite-state MDP $M = (\mathcal{H}/_\sim, A, T_H, R_H, \gamma)$ where the states are given by the set of equivalence classes, the actions and discount factor from the POMDP, and transition and reward functions defined as*

$$T_H([haz]_\sim \mid [h]_\sim, a) = \sum_{s \in S} b(s \mid [h]_\sim) \sum_{s' \in S} T(s' \mid s, a) O(z \mid s', a),$$
$$R_H([h]_\sim, a) = \sum_{s \in S} b(s \mid [h]_\sim) R(s, a).$$

Under bisimulation equivalence, the finite-history MDP and the POMDP are related in the following fundamental way.

Theorem 1 (Optimal finite-memory policies under bisimilarity)**.** *An optimal policy π^* in the finite-history MDP is an optimal finite-memory policy for the POMDP.*

Theorem 1 is a direct result of bisimilarity (Givan, Dean, and Greig 2003). We may number the equivalence classes in the finite-history MDP in such a way that they correspond to memory nodes of an FSC. As a result, the finite-history MDP can be defined on a state space consisting of memory nodes and observations rather than histories.

Definition 7 (Finite-history MDP via FSC)**.** *A POMDP satisfying Assumption 1 is a fully observable finite-state MDP $M = (\mathcal{N} \times Z, A, T_H, R_H, \gamma)$ where the states are given by pairs of memory nodes from an FSC and observations, the actions from the POMDP, and transition and reward functions defined as*

$$T_H(\langle n', z' \rangle \mid \langle n, z \rangle, a) = \sum_{s \in S} b(s \mid \langle n, z \rangle) \sum_{s' \in S} T(s' \mid s, a) O(z' \mid s', a) \eta(n' \mid n, z', a),$$
$$R_H(\langle n, z \rangle, a) = \sum_{s \in S} b(s \mid \langle n, z \rangle) R(s, a),$$

where $b(s \mid \langle n, z \rangle)$ is the belief of being in state s of the POMDP, given memory node n and observation z.

Recall that η is the memory update function of the FSC. This finite-history MDP will serve as the (unknown) true MDP M^* in our application of SPIBB.

3.2 Estimating the Finite-History MDP

Next, we describe how to estimate the true finite-history MDP M^* by an MLE-MDP $\tilde{M}$. The approach is similar to that of SPI for MDPs described in Section 2, except that the dataset $\mathcal{D}$ is different. Here, $\mathcal{D}$ is collected from simulating the POMDP $\mathcal{M}$ under (FSC) policy π_β. This yields a dataset of the form

$$\mathcal{D} = \langle \langle n_t, z_t \rangle, a_t, r_t \rangle_{t \in [1:m]}, \tag{3}$$

where the observations z_t come from the observation function, and the memory nodes n_t are observed from the FSC.

Definition 8 (Finite-history MLE-MDP)**.** *The MDP from Definition 7 can be estimated from a dataset $\mathcal{D}$ of the form* (3)*, following the same approach for estimating a standard MLE-MDP as in Definition 4:*

$$\tilde{T}_H(\langle n', z' \rangle \mid \langle n, z \rangle, a) = \frac{\#_{\mathcal{D}}(\langle n, z \rangle, a, \langle n', z' \rangle)}{\#_{\mathcal{D}}(\langle n, z \rangle, a)}, \text{ and}$$
$$\tilde{R}_H(\langle n, z \rangle, a) = \frac{R_{total}(\langle n, z \rangle, a)}{\#_{\mathcal{D}}(\langle n, z \rangle, a)}, \text{ where}$$
$$R_{total}(\langle n, z \rangle, a) = \sum_{(\langle n_t, z_t \rangle, a_t, r_t) \in \mathcal{D}} \mathbb{I}(\langle n_t, z_t \rangle = \langle n, z \rangle \wedge a_t = a) \cdot r_t.$$

3.3 Applying SPIBB to the Finite-History MDP

In this section, we apply the theory of SPIBB, as introduced in Section 2, to our setting. In particular, we have just defined

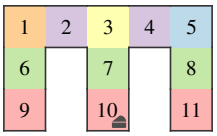

Figure 2: The Maze environment. The locations are colored according to the agent's perception.

a true MDP M^* (the finite-history MDP, Definition 6) and an MLE-MDP $\tilde{M}$ estimating M^* (Definition 8). Let

$$\mathcal{U} = \{(\langle n, z\rangle, a) \in \mathcal{N} \times Z \times A \mid \#_{\mathcal{D}}(\langle n, z\rangle, a) \leq N_\wedge\}$$

be the set of tuples $(\langle n, z\rangle, a)$ which occur less than $N_\wedge$ times in the dataset $\mathcal{D}$ for some hyperparameter $N_\wedge$. Just as in SPIBB for MDPs, we compute a new policy $\pi_I \in \Pi_k^\eta$ for the MLE-MDP $\tilde{M}$ that estimates the finite-history MDP, constrained to follow the behavior policy π_β used to collect $\mathcal{D}$ for all $(\langle n, z\rangle, a) \in \mathcal{U}$.

Theorem 2 (ζ-bound on history MDP)**.** *Let Π_β be the set of policies under the constraint of following π_β when $(\langle n, z\rangle, a) \in \mathcal{U}$. Then, the policy π_I computed by the SPIBB algorithm on the history MDP (Definition 2) is a ζ-approximate safe policy improvement over the behavior policy π_β with high probability $1 - \delta$, where:*

$$\zeta = \frac{4V_{\max}}{1-\gamma}\sqrt{\frac{2}{N_\wedge}\log\frac{2|\mathcal{H}||A|2^{|Z|}}{\delta}} - \rho(\pi_I, \tilde{M}) + \rho(\pi_\beta, \tilde{M}).$$

The proof replaces the regular MDP from the SPIBB algorithm with the (infinite) history MDP. We can reduce the exponent from $|\mathcal{H}|$, which would be the result of naively applying the SPIBB algorithm, to $|Z|$ because of the structure of the transition function of the history MDP. In particular, the transition function of the history MDP is defined for histories h which are appended by an action a and an observation z to haz, see Definition 2. As such, the successor states of h in the history MDP are fully determined by the observation z instead of the full state-space, and thus we may replace $2^{|S|}$ from Equation (1) by $2^{|Z|}$. The full proof can be found in Appendix A (Simão, Suilen, and Jansen 2023).

While Theorem 2 and its proof reason over the full history MDP, these results extend to the finite-history MDP when Assumption 1 is satisfied. We have the following corollary.

Corollary 1 (ζ-bound on finite-history MDP)**.** *Let Π_β and π_β be as in Theorem 2. Then, the policy π_I computed by the SPIBB algorithm in the finite-history MDP M^* of a POMDP satisfying Assumption 1 is a ζ-approximate safe policy improvement over the behavior policy π_β with high probability $1 - \delta$, where the admissible performance loss ζ is given by*

$$\zeta = \frac{4V_{\max}}{1-\gamma}\sqrt{\frac{2}{N_\wedge}\log\frac{2|\mathcal{N}\times Z||A|2^{|Z|}}{\delta}} - \rho(\pi_I, \tilde{M}) + \rho(\pi_\beta, \tilde{M}).$$

Since bisimilarity is an equivalence relation, the finite-history MDP is equivalent to the full history MDP, and thus also the POMDP, see Theorem 1. As a consequence, the proof of Corollary 1 follows immediately from Theorem 2 and the fact that bisimulation is an equivalence relation.

4 Empirical Analysis

This section contains the empirical evaluation of our approach to SPI for POMDPs. We first describe the setup of the experiments and then present and analyze the results. We provide further details in Appendix B (Simão, Suilen, and Jansen 2023) and code at https://github.com/LAVA-LAB/spi_pomdp.

4.1 Setup

Environments. We consider three POMDP problems:

i) CheeseMaze (McCallum 1993): An agent navigates a maze, moving in the four cardinal directions, but in each state it only perceives whether or not there is a barrier in each direction (see Figure 2). The agent is placed at a random location at the beginning of an episode, and receives a positive reward ($+1$) if it reaches the goal (⏏), and a small negative reward (-0.01) otherwise. The episode ends when the agent reaches the goal.

ii) Tiger (Kaelbling, Littman, and Cassandra 1998): An agent is in front of two doors, and a tiger is randomly positioned behind one of them at the beginning of each episode. The agent has three actions: Listening, or opening one of the doors. Listening gives a noisy observation of the position of the tiger, and a small negative reward (-1). Opening the door with the tiger gives a large negative reward (-100), while opening the other door gives a positive reward ($+10$).

iii) Voicemail (Williams and Young 2007): An agent controls a voicemail machine, at the beginning of the episode, the user listens to a message and decides if they want to keep it. This information is hidden from the agent, which has three actions: *ask*, *save*, and *delete*. Asking the user if they want to keep the message gives the agent a small negative reward (-1) and a noisy observation of the user's intention. Correctly saving the message gives a positive reward ($+5$), and a negative reward (-10) otherwise. Correctly deleting the message gives a positive reward ($+5$), and a negative reward (-20) otherwise.

Satisfaction of Assumption 1. Note that the Maze environment is close to satisfying Assumption 1 for memory that looks back two steps, *i.e.*, $k = 2$, with the exceptions of histories with equal observations. Tiger and Voicemail do not satisfy the assumption for any k.

Data collection. We generate behavior policies via Q-learning using the memory of an FSC that keeps track of the last $k \in \{1, 2\}$ observations as the state. After convergence, we extract a softmax policy, to ensure we sample different actions during data collection. We consider datasets of different sizes, namely: $1, 2, 5, 10, 20, 50, \cdots, 5\,000$, and $10\,000$ trajectories, and generate 500 datasets for each environment, number of trajectories, and behavior policy.

Learning. We consider two algorithms to compute a new policy: SPIBB, and Basic RL. Both algorithms operate on the finite-history MLE-MDP (Definition 8) related to the finite-history MDP of the POMDP. We implement Basic RL as an unconstrained SPIBB where $N_\wedge = 0$, that is,

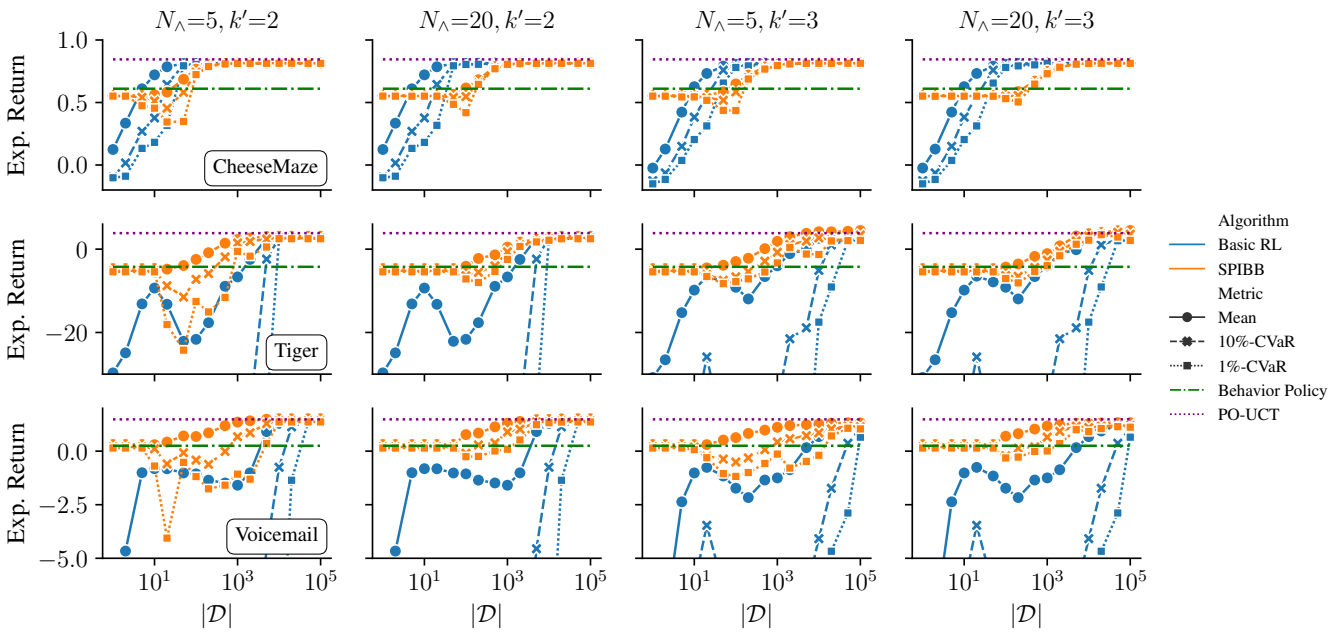

Figure 3: Policy improvement on the environments Maze, Tiger, and Voicemail (first, second and third row, respectively) for datasets collected by a behavior policy with history size $k = 2$, varying the hyperparameters pairs column-wise: $(N_\wedge = 5, k' = 2)$, $(N_\wedge = 20, k' = 2)$, $(N_\wedge = 5, k' = 3)$, and $(N_\wedge = 20, k' = 3)$. The plots show the mean (solid line), 10%-CVaR (dashed line), and 1%-CVaR (dotted line). The performance of the behavior policy is shown in green (dash-dotted line).

it solves the MLE-MDP using value iteration. For each dataset, we compute new policies π_I using each offline RL algorithm, considering different hyperparameters: $N_\wedge \in \{5, 7, 10, 15, 20, 30, 50, 70, 100\}$ and $k' \in \{k, k+1\}$, where k' is the history size encoded in the FSC of π_I.

Evaluation metrics. Each policy is evaluated over 10 000 episodes to obtain an estimate of the performance of the improved policy $\rho(\pi_I, M^*)$. We also consider the normalized policy improvement:

$$\bar{\rho}(\pi_I) = \frac{\rho(\pi_I, M^*) - \rho(\pi_\beta, M^*)}{\rho(\pi_{\max}, M^*) - \rho(\pi_\beta, M^*)},$$

where $\pi_{\max}$ is the policy with the highest expected return in each environment. To aggregate the results across the 500 repetitions, we compute the mean and Conditional Value at Risk (CVaR; Rockafellar and Uryasev 2000). We use x%-CVaR to indicate the mean of the x% lowest performances. As an approximation of the optimal value, we show the performance of PO-UCT (Silver and Veness 2010), which uses the environment as a simulator to compute a policy.

4.2 Results

Figure 3 shows results on the three environments (ordered by row). The data was collected using a behavior policy with $k = 2$. The first column shows the results where SPIBB uses a low threshold to consider a history-action pair known and the same memory size as the behavior policy ($N_\wedge = 5$ and $k' = 2$). The second column shows the results with a higher threshold ($N_\wedge = 20$ and $k' = 2$). The third column shows the results for increased memory ($N_\wedge = 5$ and $k' = 3$). Finally, the fourth column shows the results with a higher threshold and increased memory ($N_\wedge = 20$ and $k' = 3$). Basic RL is included everywhere to give a perspective on the influence of different hyperparameters.

Figures 4 and 5 extend the empirical analysis on the Voicemail and Tiger environments for memoryless behavior policy ($k = 1$), since they demonstrated to be more challenging for the safe policy improvement problem. Figure 4 considers the Voicemail environment, while Figure 5 shows the normalized results for a range of thresholds in the Tiger environment. We provide further results in Appendix C (Simão, Suilen, and Jansen 2023).

4.3 Analysis

Basic RL is unreliable. Across all environments, the Basic RL algorithm shows a considerable performance drop compared to the behavior policy, even in terms of the mean performance for smaller datasets. Notice that for Tiger and Voicemail, the CVaR metrics are often outside the graph.

SPIBB outperforms Basic RL. In the environments Tiger and Voicemail (Figure 3, second and third row), the SPIBB algorithm shows better performance than the Basic RL across all dataset sizes. This is likely due to the SPIBB algorithm retaining the randomization of the behavior policy when insufficient data is available.

SPIBB is reliable when Assumption 1 is satisfied. Analyzing the results for the Maze environment (Figure 3, first row), we observe that SPIBB shows reliably outperforms

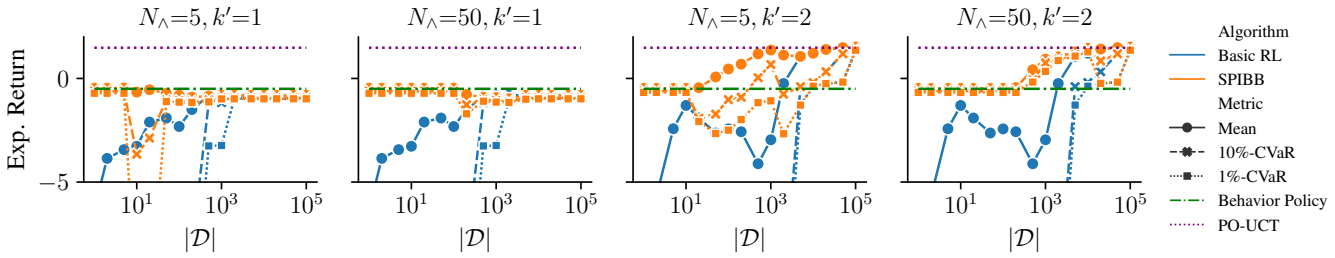

Figure 4: Policy improvement on the Voicemail environment for datasets collected with a memoryless policy ($k = 1$), varying the hyperparameters pairs column-wise: $(N_\wedge = 5, k' = 1)$, $(N_\wedge = 50, k' = 1)$, $(N_\wedge = 5, k' = 2)$, and $(N_\wedge = 50, k' = 2)$. The plots show the mean (solid line), 10%-CVaR (dashed line) and 1%-CVaR (dotted line). The performance of the behavior policy is shown in green (dash-dotted line).

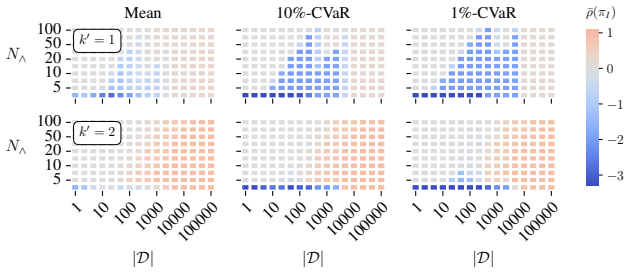

Figure 5: Normalized performance $\bar{\rho}(\pi_I)$ on the Tiger environment ($k = 1$). The left, middle and right columns show the mean, 10%-CVaR and 1%-CVaR, respectively. The first row shows the results where the improved policy uses the same memory as the behavior policy ($k' = k$), while the second row shows the results for an improved policy with more memory ($k' = k + 1$).

the behavior policy even for a small $N_\wedge$ (first column), for which only the 1%-CVaR shows a performance drop.

More memory improves the reliability. SPIBB shows slightly unreliable behavior for small values of $N_\wedge$ in the Tiger and Voicemail environments (Figure 3), as evidenced by both the CVaR curves, which can be alleviated by increasing the $N_\wedge$ or the memory of the new policy (second, third and fourth column). When Assumption 1 is violated, the performance drop may be significant, as seen in the first two columns of Figure 4. In this case, merely increasing the $N_\wedge$ threshold is not enough to guarantee a policy improvement. Increasing the memory size, however, allows the SPIBB algorithm to improve the behavior policy, as Figure 4 (last column) and Figure 5 (second row) show.

Deterministic policies may require more memory. Figure 4 shows an interesting phenomenon. In partially observable settings, the stochastic behavior policy might perform better than the new deterministic policy, since randomization can trade-off some amount of memory. We observe that when $k = 1$, SPIBB and Basic RL converge to deterministic policies with an expected return lower than the behavior policy. When SPIBB has sufficient data, it is not constrained to follow the behavior policy, and thus does not inherit any randomization from that policy. As stated in the previous paragraph, more memory can then yield a new deterministic policy with a higher return than the behavior policy.

5 Related Work

Offline RL, also known as batch RL, learns or evaluates a policy from a fixed batch of historical data (Levine et al. 2020). Overall, these algorithms rely on pessimism to mitigate the lack of feedback from the environment and can be split into two categories (Jin, Yang, and Wang 2021): those that constrain the final policy to stay close to the behavior policy (Laroche, Trichelair, and des Combes 2019), and those that penalize rare experiences (Petrik, Ghavamzadeh, and Chow 2016). Our method belongs to the first category.

Various extensions of SPIBB could be adapted for POMDPs, such as soft-SPIBB (Nadjahi, Laroche, and des Combes 2019; Scholl et al. 2022), deep-SPIBB (Brandfonbrener, des Combes, and Laroche 2022), and factored-SPIBB (Simão and Spaan 2019a,b). SPI has also been studied without the behavior policy (Simão, Laroche, and des Combes 2020) and for multi-objective (Satija et al. 2021) and non-stationary settings (Chandak et al. 2020).

When the behavior policy is influenced by unobserved variables, we may come across confounding variables. The problem of evaluating a policy offline was studied in this setting, for instance, assuming that observed and unobserved variables are decoupled (Tennenholtz, Shalit, and Mannor 2020), or that the influence of the confounding variable on the behavior policy is limited (Namkoong et al. 2020). Since we assume that the behavior policy only depends on the observed history, we have no confounding variables.

6 Conclusions

We presented a new approach to safe policy improvement for POMDPs. Our experiments show the applicability of the approach, even in cases where finite-history is not sufficient to obtain optimal results. In the future, it would be interesting to relax Assumption 1 to distance metrics (Ferns, Panangaden, and Precup 2004, 2005) instead of exact bisimilarity.

Acknowledgments

We would like to thank Alberto Castellini, Alessandro Farinelli, Matthijs Spaan, and Edoardo Zorzi for discussions on related topics. We also thank Maris Galesloot for help with the implementation of the PO-UCT algorithm. This research has been partially funded by NWO grants OCENW.KLEIN.187 (Provably Correct Policies for Uncertain Partially Observable Markov Decision Processes), NWA.1160.18.238 (PrimaVera), and the ERC Starting Grant 101077178 (DEUCE).

References

Amato, C.; Bernstein, D. S.; and Zilberstein, S. 2010. Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *Auton. Agents Multi Agent Syst.*, 21(3): 293–320.

Andriotis, C. P.; and Papakonstantinou, K. G. 2021. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliab. Eng. Syst. Saf.*, 212: 107551.

Åström, K. J. 1965. Optimal control of Markov processes with incomplete state information. *Journal of mathematical analysis and applications*, 10(1): 174–205.

Bonet, B. 2002. An epsilon-Optimal Grid-Based Algorithm for Partially Observable Markov Decision Processes. In *ICML*, 51–58. Morgan Kaufmann.

Brandfonbrener, D.; des Combes, R. T.; and Laroche, R. 2022. Incorporating Explicit Uncertainty Estimates into Deep Offline Reinforcement Learning. *arXiv preprint arXiv:2206.01085*.

Carr, S.; Jansen, N.; and Topcu, U. 2021. Task-Aware Verifiable RNN-Based Policies for Partially Observable Markov Decision Processes. *J. Artif. Intell. Res.*, 72: 819–847.

Chadès, I.; Carwardine, J.; Martin, T. G.; Nicol, S.; Sabbadin, R.; and Buffet, O. 2012. MOMDPs: A Solution for Modelling Adaptive Management Problems. In *AAAI*, 267–273. AAAI Press.

Chadès, I.; Martin, T. G.; Nicol, S.; Burgman, M. A.; Possingham, H. P.; and Buckley, Y. M. 2011. General rules for managing and surveying networks of pests, diseases, and endangered species. *Proceedings of the National Academy of Sciences*, 108(20): 8323–8328.

Chandak, Y.; Jordan, S. M.; Theocharous, G.; White, M.; and Thomas, P. S. 2020. Towards Safe Policy Improvement for Non-Stationary MDPs. In *NeurIPS*, 9156–9168.

Cheng, C.; Xie, T.; Jiang, N.; and Agarwal, A. 2022. Adversarially Trained Actor Critic for Offline Reinforcement Learning. In *ICML*, volume 162, 3852–3878. PMLR.

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

Dujardin, Y.; Dietterich, T.; and Chadès, I. 2017. Three New Algorithms to Solve N-POMDPs. In *AAAI*, 4495–4501. AAAI Press.

Ferns, N.; Panangaden, P.; and Precup, D. 2004. Metrics for Finite Markov Decision Processes. In *UAI*, 162–169. AUAI Press.

Ferns, N.; Panangaden, P.; and Precup, D. 2005. Metrics for Markov Decision Processes with Infinite State Spaces. In *UAI*, 201–208. AUAI Press.

Givan, R.; Dean, T. L.; and Greig, M. 2003. Equivalence notions and model minimization in Markov decision processes. *Artif. Intell.*, 147(1-2): 163–223.

Jin, Y.; Yang, Z.; and Wang, Z. 2021. Is Pessimism Provably Efficient for Offline RL? In *ICML*, volume 139, 5084–5096. PMLR.

Junges, S.; Jansen, N.; Wimmer, R.; Quatmann, T.; Winterer, L.; Katoen, J.; and Becker, B. 2018. Finite-State Controllers of POMDPs using Parameter Synthesis. In *UAI*, 519–529. AUAI Press.

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

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement Learning: A Survey. *J. Artif. Intell. Res.*, 4: 237–285.

Kochenderfer, M. J. 2015. *Decision making under uncertainty: theory and application*. MIT press.

Laroche, R.; Trichelair, P.; and des Combes, R. T. 2019. Safe Policy Improvement with Baseline Bootstrapping. In *ICML*, volume 97, 3652–3661. PMLR.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643*.

Madani, O.; Hanks, S.; and Condon, A. 2003. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artif. Intell.*, 147(1-2): 5–34.

McCallum, A. 1993. Overcoming Incomplete Perception with Utile Distinction Memory. In *ICML*, 190–196. Morgan Kaufmann.

Meuleau, N.; Kim, K.; Kaelbling, L. P.; and Cassandra, A. R. 1999a. Solving POMDPs by Searching the Space of Finite Policies. In *UAI*, 417–426. Morgan Kaufmann.

Meuleau, N.; Peshkin, L.; Kim, K.; and Kaelbling, L. P. 1999b. Learning Finite-State Controllers for Partially Observable Environments. In *UAI*, 427–436. Morgan Kaufmann.

Nadjahi, K.; Laroche, R.; and des Combes, R. T. 2019. Safe Policy Improvement with Soft Baseline Bootstrapping. In *ECML/PKDD (3)*, volume 11908, 53–68. Springer.

Namkoong, H.; Keramati, R.; Yadlowsky, S.; and Brunskill, E. 2020. Off-policy Policy Evaluation For Sequential Decisions Under Unobserved Confounding. In *NeurIPS*, 18819–18831.

Petrik, M.; Ghavamzadeh, M.; and Chow, Y. 2016. Safe Policy Improvement by Minimizing Robust Baseline Regret. In *NIPS*, 2298–2306.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1st edition.

Rockafellar, R. T.; and Uryasev, S. 2000. Optimization of conditional value-at-risk. *Journal of risk*, 2(3): 21–41.

Satija, H.; Thomas, P. S.; Pineau, J.; and Laroche, R. 2021. Multi-Objective SPIBB: Seldonian Offline Policy Improvement with Safety Constraints in Finite MDPs. In *NeurIPS*, 2004–2017.

Scholl, P.; Dietrich, F.; Otte, C.; and Udluft, S. 2022. Safe Policy Improvement Approaches on Discrete Markov Decision Processes. In *ICAART (2)*, 142–151. SCITEPRESS.

Silver, D.; and Veness, J. 2010. Monte-Carlo Planning in Large POMDPs. In *NIPS*, 2164–2172. Curran Associates, Inc.

Simão, T. D.; Laroche, R.; and des Combes, R. T. 2020. Safe Policy Improvement with an Estimated Baseline Policy. In *AAMAS*, 1269–1277. IFAAMAS.

Simão, T. D.; and Spaan, M. T. J. 2019a. Safe Policy Improvement with Baseline Bootstrapping in Factored Environments. In *AAAI*, 4967–4974. AAAI Press.

Simão, T. D.; and Spaan, M. T. J. 2019b. Structure Learning for Safe Policy Improvement. In *IJCAI*, 3453–3459. ijcai.org.

Simão, T. D.; Suilen, M.; and Jansen, N. 2023. Safe Policy Improvement for POMDPs via Finite-State Controllers. *arXiv preprint arXiv:2301.04939*.

Smallwood, R. D.; and Sondik, E. J. 1973. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Oper. Res.*, 21(5): 1071–1088.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement Learning — An Introduction*. Adaptive computation and machine learning. MIT Press.

Tennenholtz, G.; Shalit, U.; and Mannor, S. 2020. Off-Policy Evaluation in Partially Observable Environments. In *AAAI*, 10276–10283. AAAI Press.

Thomas, P. S.; Theocharous, G.; and Ghavamzadeh, M. 2015. High Confidence Policy Improvement. In *ICML*, volume 37, 2380–2388. JMLR.org.

Weissman, T.; Ordentlich, E.; Seroussi, G.; Verdu, S.; and Weinberger, M. J. 2003. Inequalities for the L_1 Deviation of the Empirical Distribution. Technical report, Hewlett-Packard Labs, Palo Alto, United States.

Williams, J. D.; and Young, S. J. 2007. Partially observable Markov decision processes for spoken dialog systems. *Comput. Speech Lang.*, 21(2): 393–422.

Yeager, J.; Moss, J. E. B.; Norrish, M.; and Thomas, P. S. 2022. Mechanizing Soundness of Off-Policy Evaluation. In *ITP*, volume 237, 32:1–32:20.