More for Less: Safe Policy Improvement With Stronger Performance Guarantees

Patrick Wienhöft[1,2], **Marnix Suilen**[3], **Thiago D. Simão**[3], **Clemens Dubslaff**[4,2], **Christel Baier**[1,2] and **Nils Jansen**[3]
[1]Department of Computer Science, Technische Universität Dresden, Dresden, Germany
[2]Centre for Tactile Internet with Human-in-the-Loop (CeTI)
[3]Department of Software Science, Radboud University, Nijmegen, The Netherlands
[4]Eindhoven University of Technology, Eindhoven, The Netherlands
{patrick.wienhoeft, christel.baier}@tu-dresden.de
{m.suilen, t.simao, n.jansen}@science.ru.nl
c.dubslaff@tue.nl

Abstract

In an offline reinforcement learning setting, the safe policy improvement (SPI) problem aims to improve the performance of a behavior policy according to which sample data has been generated. State-of-the-art approaches to SPI require a high number of samples to provide practical probabilistic guarantees on the improved policy's performance. We present a novel approach to the SPI problem that provides the means to require less data for such guarantees. Specifically, to prove the correctness of these guarantees, we devise implicit transformations on the data set and the underlying environment model that serve as theoretical foundations to derive tighter improvement bounds for SPI. Our empirical evaluation, using the well-established SPI with baseline bootstrapping (SPIBB) algorithm, on standard benchmarks shows that our method indeed significantly reduces the sample complexity of the SPIBB algorithm.

1 Introduction

Markov decision processes (MDPs) are the standard model for sequential decision-making under uncertainty [Puterman, 1994]. *Reinforcement learning* (RL) solves such decision-making problems, in particular when the environment dynamics are unknown [Sutton and Barto, 1998].

In an *online* RL setting, an agent aims to learn a decision-making policy that maximizes the expected accumulated reward by interacting with the environment and observing feedback, typically in the form of information about the environment state and reward. While online RL has shown great performance in solving hard problems [Mnih *et al.*, 2015; Silver *et al.*, 2018], the assumption that the agent can always directly interact with the environment is not always realistic. In real-world applications such as robotics or healthcare, direct interaction can be impractical or dangerous [Levine *et al.*, 2020]. Furthermore, alternatives such as simulators or digital twins may not be available or insufficiently capture the nuances of the real-world application for reliable learning [Ramakrishnan *et al.*, 2020; Zhao *et al.*, 2020].

Offline RL (or batch RL) [Lange *et al.*, 2012] mitigates this concern by restricting the agent to have only access to a fixed data set of past interactions. As a common assumption, the data set has been generated by a so-called *behavior policy*. An offline RL algorithm aims to produce a new policy without further interactions with the environment [Levine *et al.*, 2020]. Methods that can reliably improve the performance of a policy are key in (offline) RL.

Safe policy improvement (SPI) algorithms address this challenge by providing (probabilistic) correctness guarantees on the reliable improvement of policies [Thomas *et al.*, 2015; Petrik *et al.*, 2016]. These guarantees depend on the size of the data set and usually adhere to a conservative bound on the minimal amount of samples required. Since this bound often turns out to be too large for practical applications of SPI, it is instead turned into a hyperparameter (see, *e.g.*, [Laroche *et al.*, 2019]). The offline nature of SPI prevents further data collection, which steers the key requirements of SPI in practical settings: (1) exploit the data set as efficiently as possible and (2) compute better policies from smaller data sets.

1.1 Contributions

Our contribution provides the theoretical foundations to improve the understanding of SPI algorithms in general. Specifically, in a general SPI setting, we can guarantee a higher performance for significantly less data. Equivalently, we can allow the same amount of data and consequently provide significantly less performance guarantees. Our main technical contribution is an transformation of the underlying MDP model into a *two-successor MDP* (2sMDP) along with adjustments to the data set, that allows us to prove these tighter bounds. A 2sMDP is an MDP where each state-action pair has at most two successors, hence limiting the branching factor of an MDP to only two. These transformations preserve (the optimal) performance of policies and are reversible. In the context of SPI these transformations are implicit, *i.e.*, do not have to be computed explicitly. Hence, we are able to apply standard SPI algorithms such as SPI with baseline bootstrapping

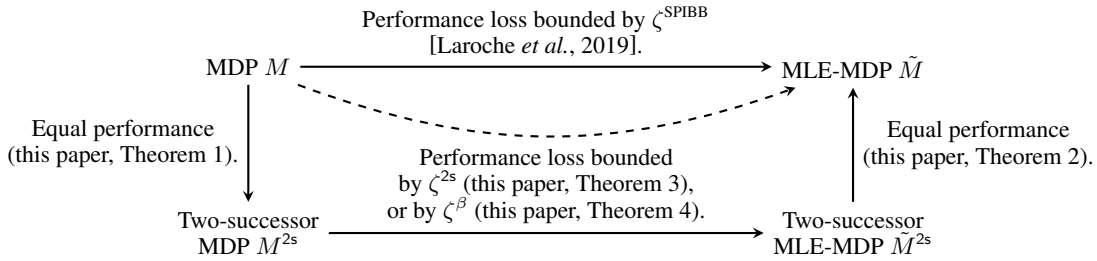


Figure 1: Overview of our approach. The solid arrows indicate how the full derivation of the improvement guarantees is done, while the dashed line indicates that the transformations are only used in the proofs and that in practice, we can immediately use ζ^{2s} or ζ^{β} as bounds.

(SPIBB) [Laroche *et al.*, 2019], and use our novel improvement guarantees without any algorithmic changes necessary, as also illustrated in Figure 1.

Following the theoretical foundations for the MDP and data set transformations (Section 4), we present two different methods to compute the new performance guarantees (Section 5). The first uses Weissman's bound [Weissman *et al.*, 2003], as also used in, *e.g.*, standard SPIBB, while the second uses the inverse incomplete beta function [Temme, 1992]. Our experimental results show a significant reduction in the amount of data required for equal performance guarantees (Section 6). Concretely, where the number of samples required at each state-action pair of standard SPIBB grows linearly in the number of states, our approach only grows logarithmic in the number of states for both methods. We also demonstrate the impact on three well-known benchmarks in practice by comparing them with standard SPIBB across multiple hyperparameters.

2 Preliminaries

Let X be a finite set. We denote the number of elements in X by $|X|$. A discrete probability distribution over X is a function $\mu\colon X \to [0,1]$ where $\sum_{x\in X}\mu(x) = 1$. The set of all such distributions is denoted by $\Delta(X)$. The L_1-distance between two probability distributions μ and σ is defined as $\|\mu-\sigma\|_1 = \sum_{x\in X}|\mu(x)-\sigma(x)|$. We write $[m:n]$ for the set of natural numbers $\{m,\dots,n\} \subset \mathbb{N}$, and $\mathbb{I}[x{=}x']$ for the indicator function, returning 1 if $x = x'$ and 0 otherwise.

Definition 1 (MDP)**.** *A* Markov decision process *(MDP) is a tuple $M = (S, A, \iota, P, R, \gamma)$, where S and A are finite sets of states and actions, respectively, $\iota \in S$ an initial state, $P\colon S \times A \rightharpoonup \Delta(S)$ is the (partial) transition function, $R\colon S \times A \rightharpoonup [-R_{max}, R_{max}]$ is the reward function bounded by some known value $R_{max} \in \mathbb{R}$, and $\gamma \in (0,1) \subset \mathbb{R}$ is the discount factor.*

We say that an action a is *enabled* in state s if $P(s,a)$ is defined. We write $P(s' \mid s,a)$ for the transition probability $P(s,a)(s')$, and $Post_M(s,a)$ for the set of successor states reachable with positive probability from the state-action pair (s,a) in M. A *path* in M is a finite sequence $\langle s_1, a_1, \dots, a_{n-1}, s_n\rangle \in (S \times A)^* \times S$ where $s_i \in Post_M(s_{i-1}, a_{i-1})$ for all $i \in [2{:}n]$. The probability of following a path $\langle s_1, a_1, \dots, a_{n-1}, s_n\rangle$ in the MDP M given a deterministic sequence of actions is written as $\mathbb{P}_M(\langle s_1, a_1, \dots, a_{n-1}, s_n\rangle)$ and can be computed by repeatedly applying the transition probability function, *i.e.*, $\mathbb{P}_M(\langle s_1, a_1, \dots, a_{n-1}, s_n\rangle) = \prod_{i=1}^{n-1} P(s_{i+1} \mid s_i, a_i)$.

A memoryless stochastic *policy* for M is a function $\pi\colon S \to \Delta(A)$. The set of such policies is Π. The goal is to find a policy maximizing the expected discounted reward

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

where r_t is the reward the agent collects at time t when following policy π in the MDP.

We write $V_M^\pi(s)$ for the state-based *value function* of an MDP M under a policy π. Whenever clear from context, we omit M and π. The value of a state s in an MDP M is the least solution of the Bellman equation and can be computed by, *e.g.*, value iteration [Puterman, 1994]. The *performance* $\rho(\pi, M)$ of a policy π in an MDP M is defined as the value in the initial state $\iota \in S$, *i.e.*, $\rho(\pi, M) = V_M^\pi(\iota)$.

3 Safe Policy Improvement

In *safe policy improvement* (SPI), we are given an MDP M with an unknown transition function, a policy π_b, also known as the *behavior policy*, and a data set $\mathcal{D}$ of paths in M under π_b. The goal is to derive a policy π_I from π_b and $\mathcal{D}$ that with high probability $1{-}\delta$ guarantees to *improve* π_b on M up to an *admissible performance loss* ζ. That is, the performance of π_I is at least that of π_b tolerating an error of ζ:

$$\rho(\pi_I, M) \geq \rho(\pi_b, M) - \zeta. \tag{1}$$

3.1 Maximum Likelihood Estimation

We use *maximum likelihood estimation* (MLE) to derive an MLE-MDP $\tilde{M}$ from the data set $\mathcal{D}$. For a path $\rho \in \mathcal{D}$, let $\#_\rho(s,a)$ and $\#_\rho(s,a,s')$ be the number of (sequential) occurrences of a state-action pair (s,a) and a transition (s,a,s') in ρ, respectively. We lift this notation the level of the data set $\mathcal{D}$ by defining $\#_\mathcal{D}(s,a) = \sum_{\rho\in\mathcal{D}} \#_\rho(s,a)$ and $\#_\mathcal{D}(s,a,s') = \sum_{\rho\in\mathcal{D}} \#_\rho(s,a,s')$.

Definition 2 (MLE-MDP)**.** *The* maximum likelihood MDP *(MLE-MDP) of an MDP $M = (S, A, \iota, P, R, \gamma)$ and data set $\mathcal{D}$ is a tuple $\tilde{M} = (S, A, \iota, \tilde{P}, R, \gamma)$ where S, ι, A, R, and γ are as in M and the transition function is estimated from $\mathcal{D}$:*

$$\tilde{P}(s' \mid s,a) = \frac{\#_\mathcal{D}(s,a,s')}{\#_\mathcal{D}(s,a)}.$$

Let $e\colon S\times A \to \mathbb{R}$ be an error function. We define $\Xi_e^{\tilde{M}}$ as the set of MDPs M' that are close to $\tilde{M}$, *i.e.*, where for all state-action pairs (s,a) the L_1-distance between the transition function $P'(\cdot \mid s,a)$ and $\tilde{P}(\cdot \mid s,a)$ is at most $e(s,a)$:

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

SPI methods aim at defining the error function e in such a way that $\Xi_e^{\tilde{M}}$ contains the true MDP M with high probability $1-\delta$. Computing a policy that is an improvement over the behavior policy for all MDPs in this set then also guarantees an improved policy for the MDP M with high probability $1-\delta$ [Petrik *et al.*, 2016]. The amount of data required to achieve a ζ^{SPI}-approximately safe policy improvement with probability $1-\delta$ (recall Equation (1)) for all state-action pairs has been established by Laroche *et al.* ([2019]) as

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

Intuitively, if the data set $\mathcal{D}$ satisfies the constraint in Equation 2, the MLE-MDP estimated from $\mathcal{D}$ will be close enough to the unknown MDP M used to obtain $\mathcal{D}$. To this end, it would be likely that a policy in the MLE-MDP with better performance will also have a better performance in M.

3.2 SPI with Baseline Bootstrapping

The constraint in Equation (2) has to hold for all state-action pairs in order to guarantee a ζ-approximate improvement and thus requires a large data set with good coverage of the entire model. SPI with baseline bootstrapping (SPIBB) [Laroche *et al.*, 2019] relaxes this requirement by only changing the behavior policy in those pairs for which the data set contains enough samples and follows the behavior policy otherwise. Specifically, state-action pairs with less than $N_\wedge^{\text{SPIBB}}$ samples are collected in a set of *unknown* state-action pairs $\mathcal{U}$:

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

SPIBB then determines an improved policy π_I similar to (standard) SPI, except that if $(s,a) \in \mathcal{U}$, π_I is required to follow the behavior policy π_b:

$$\forall (s,a) \in \mathcal{U}.\, \pi_I(a \mid s) = \pi_b(a \mid s).$$

Then, π_I is an improved policy as in Equation (1), where $N_\wedge^{\text{SPIBB}}$ is treated as a hyperparameter and ζ is given by

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

We can rearrange this equation to compute the number of necessary samples for a ζ^{SPIBB}-approximate improvement. As $\rho(\pi_I,\tilde{M})$ is only known at runtime, we have to employ an under-approximation $\rho(\pi_b,\tilde{M})$ to a priori compute

$$N_\wedge^{\text{SPIBB}} = \frac{32V_{max}^2}{(\zeta^{\text{SPIBB}})^2(1-\gamma)^2}\log\frac{2|S||A|2^{|S|}}{\delta}.$$

Thus, the sample size constraint $N_\wedge^{\text{SPIBB}}$ grows approximately linearly in terms of the size of the MDP. The exponent of the term $2^{|S|}$ is an over-approximation of the maximum branching factor of the MDP, since worst-case, the MDP can be fully connected. In the following Section, we present our approach to limit the branching factor of an MDP. After that, we present two methods that exploit this limited branching factor to derive improved sampling size constraints for SPI that satisfy the same guarantees.

4 Tighter Improvement Bounds for SPI

In the following, we present the technical construction of two-successor MDPs and the data set transformation that allows us to derive the tighter performance guarantees in SPI.

4.1 From MDP to Two-Successor MDP

A *two-successor MDP* (2sMDP) is an MDP M^{2s} where each state-action pair (s,a) has at most two possible successors states, *i.e.*, $|Post_{M^{2s}}(s,a)| \leq 2$. To transform an MDP $M = (S,A,\iota,P,R,\gamma)$ into a 2sMDP, we introduce a set of *auxiliary* states S_{aux} along with the *main* states S of the MDP M. Further, we include an additional action τ and adapt the probability and reward functions towards a 2sMDP $M^{2s} = (S \cup S_{\text{aux}}, A \cup \{\tau\}, \iota, P^{2s}, R^{2s}, \gamma^{2s})$.

For readability, we now detail the transformation for a fixed state-action pair (s,a) with three or more successors. The transformation of the whole MDP follows from repeatedly applying this transformation to all such state-action pairs.

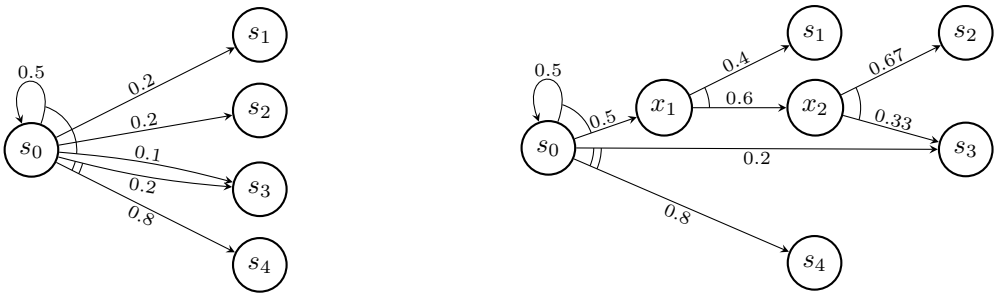
We enumerate the successor states of (s,a), *i.e.*, $Post_M(s,a) = \{s_1,\dots,s_k\}$ and define $p_i = P(s_i \mid s,a)$ for all $i = 1,\dots,k$. Further, we introduce $k-2$ auxiliary states $S_{\text{aux}}^{s,a} = \{x_2,\dots,x_{k-1}\}$, each with one available action with a binary outcome. Concretely, the two possible outcomes in state x_i are "move to state s_i" or "move to one of the states $s_{i+1},\dots,s_k$" where the latter is represented by moving to an auxiliary state x_{i+1}, unless $i = k-1$ in which case we immediately move to s_k. Formally, the new transition function $P^{2s}(\cdot \mid s,a)$ is:

$$P^{2s}(s_1 \mid s,a) = p_1, \quad P^{2s}(x_2 \mid s,a) = 1-p_1.$$

For the transition function P^{2s} in the auxiliary states we define a new action τ that will be the only enabled action in these states. For $i > 1$, the transition function P^{2s} is then

$$P^{2s}(s_i \mid x_i,\tau) = \frac{p_i}{1-(p_1+\dots+p_{i-1})},$$
$$P^{2s}(x_{i+1} \mid x_i,\tau,i<k-1) = 1 - \frac{p_i}{1-(p_1+\dots+p_{i-1})},$$
$$P^{2s}(s_k \mid x_{k-1},\tau) = 1 - \frac{p_k}{1-(p_{i-1}+p_k)}.$$

An example of this transformation is shown in Figure 2, where Figure 2a shows the original MDP and Figure 2b shows the resulting 2sMDP. As we introduce $|Post(s,a)|$ auxiliary states for a state-action pair (s,a), and $k \leq |S|$ in the worst-case of a fully connected MDP, we can bound the number of states in the 2sMDP by $|S \cup S_{\text{aux}}| \leq |S| + |S||A|(|S|-2) \leq |S|^2|A|$. Note that we did not specify a particular order for the enumeration of the successor states. Further, other transformations utilizing auxiliary states with a different structure (*e.g.*, a balanced binary tree) are possible.



(a) MDP M. (b) 2sMDP M^{2s}.

Figure 2: Example for a transformation from an MDP to a 2sMDP, where the single and double arc indicate different actions.

However, neither the structure of the auxiliary states, nor the order of successor states changes the total number of states in the 2sMDP, which is the deciding factor for the application of this transformation in the context of SPI algorithms.

The extension of the reward function is straightforward, i.e., the agent receives the same reward as in the original MDP when in main states and no reward when in auxiliary states:

$$R^{2s}(s,a) = \begin{cases} R(s,a) & s \in S, a \in A, \\ 0 & \text{otherwise.} \end{cases}$$

Any policy π for the MDP M can be extended into a policy π^{2s} for the 2sMDP M^{2s} by copying π for states in S and choosing τ otherwise:

$$\pi^{2s}(a \mid s) = \begin{cases} \pi(a \mid s) & s \in S, \\ \mathbb{I}[a = \tau] & s \in S_{\text{aux}}. \end{cases}$$

Finally, in order to preserve discounting correctly, we introduce a state-dependent discount factor γ^{2s}, such that discounting only occurs in the main states, *i.e.*,

$$\gamma^{2s}(s) = \begin{cases} \gamma & s \in S, \\ 1 & s \in S_{\text{aux}}. \end{cases}$$

This yields the following value function for the 2sMDP M^{2s}:

$$V^{\pi^{2s}}_{M^{2s}}(s) = \sum_{a \in A} \pi^{2s}(a \mid s) \Big(R^{2s}(s,a) + \gamma^{2s}(s) \sum_{s' \in S} P^{2s}(s' \mid s,a) V^{\pi^{2s}}_{M^{2s}}(s') \Big).$$

The performance of policy π^{2s} on M^{2s} uses the value function defined above and is denoted by $\rho^{2s}(\pi^{2s}, M^{2s}) = V^{\pi^{2s}}_{M^{2s}}(\iota)$, for the initial state $\iota \in S$. Our transformation described above, together with the adjusted value function, indeed preserves the performance of the original MDP and policy:

Theorem 1 (Preservation of transition probabilities)**.** *For every transition (s,a,s') in the original MDP M, there exists a unique path $\langle s, a, x_2, \tau, \ldots, x_i, \tau, s' \rangle$ in the 2sMDP M^{2s} with the same probability. That is,*

$$\mathbb{P}_M(\langle s,a,s' \rangle) = \mathbb{P}_{M^{2s}}(\langle s, a, x_2, \tau, \ldots, x_i, \tau, s' \rangle).$$

Wienhöft *et al.* [2023] prove all theoretical results.

Corollary 1 (Preservation of performance)**.** *Let M be an MDP, π a policy for M, and M^{2s} the two-successor MDP with policy π^{2s} constructed from M and π as described above. Then $\rho(\pi, M) = \rho^{2s}(\pi^{2s}, M^{2s})$.*

4.2 Data-set Transformation

In the previous section, we discussed how to transform an MDP into a 2sMDP. However, for SPI we do not have access to the underlying MDP, but only to a data set $\mathcal{D}$ and the behavior policy π_b used to collect this data. In this section, we present a transformation similar to the one from MDP to 2sMDP, but now for the data set $\mathcal{D}$. This data set transformation allows us to estimate a 2sMDP from the transformed data via maximum likelihood estimation (MLE).

We again assume a data set $\mathcal{D}$ of observed states and actions of the form $\mathcal{D} = \langle s_t, a_t \rangle_{t \in [1:m]}$ from an MDP M. We transform the data set $\mathcal{D}$ into a data set $\mathcal{D}^{2s}$ that we use to define a two-successor MLE-MDP $\tilde{M}^{2s}$. Each sample (s_t, a_t, s_{t+1}) in $\mathcal{D}$ is transformed into a set of samples, each corresponding to a path from s_t to s_{t+1} via states in S_{aux} in M^{2s}. Importantly, the data set transformation only relies on $\mathcal{D}$ and not on any additional knowledge about M.

Similar to the notation in Section 3, let $\#_{\mathcal{D}}(x)$ denote the number of times x occurs in $\mathcal{D}$. For each state-action pair $(s,a) \in S \times A$ we denote its successor states in $\tilde{M}$ as $Post_{\tilde{M}}(s,a) = \{s_i \mid \#_{\mathcal{D}}(s,a,s_i) > 0\}$, which are again enumerated by $\{s_1, \ldots, s_k\}$. Similarly as for the MDP transformation, we define $Post_{\tilde{M}^{2s}}(s,a) = Post_{\tilde{M}}(s,a)$ if $k \leq 2$ and $Post_{\tilde{M}^{2s}}(s,a) = \{s_1, x_2\}$ otherwise. For auxiliary states $x_i \in S^{s,a}_{\text{aux}}$, we define $Post_{\tilde{M}^{2s}}(x_i, \tau) = \{s_i, x_{i+1}\}$ for $i < k-1$ and $Post_{\tilde{M}^{2s}}(x_{k-1}, \tau) = \{s_{k-1}, s_k\}$. We then define the transformed data set $\mathcal{D}^{2s}$ from $\mathcal{D}$ for each $s \in S$ and $s' \in Post_{\tilde{M}^{2s}}(s,a)$ as follows:

$$\#_{\mathcal{D}^{2s}}(s,a,s') = \begin{cases} \#_{\mathcal{D}}(s,a,s') & s' \in S, \\ \sum_{j=2}^{k} \#_{\mathcal{D}}(s,a,s_j) & s' = x_2 \in S^{s,a}_{\text{aux}}, \\ 0 & \text{otherwise.} \end{cases}$$

Further, for each $x_i \in S^{s,a}_{\text{aux}}$ and $s' \in Post_{\tilde{M}^{2s}}(s,a)$

$$\#_{\mathcal{D}^{2s}}(x_i,\tau,s') = \begin{cases} \#_{\mathcal{D}}(s,a,s') & s' \in S, \\ \sum_{j=i+1}^{k} \#_{\mathcal{D}}(s,a,s_j) & s' = x_{i+1} \in S^{s,a}_{\text{aux}}, \\ 0 & \text{otherwise.} \end{cases}$$

The following preservation results for data generated MLE-MDPs are in the line of Theorem 1 and Corollary 1. See Figure 1 for an overview of the relationships between theorems.

Theorem 2 (Preservation of estimated transition probabilities)**.** *Let $\mathcal{D}$ be a data set and $\mathcal{D}^{2s}$ be the data set obtained by the transformation above. Further, let $\tilde{M}$ and $\tilde{M}^{2s}$ be the MLE-MDPs constructed from $\mathcal{D}$ and $\mathcal{D}^{2s}$, respectively. Then for every transition (s, a, s') in $\tilde{M}$ there is a unique path $\langle s, a, x_2, \tau, \ldots, x_i, \tau, s' \rangle$ in $\tilde{M}^{2s}$ with the same probability:*

$$\mathbb{P}_{\tilde{M}}(\langle s, a, s' \rangle) = \mathbb{P}_{\tilde{M}^{2s}}(\langle s, a, x_2, \tau, \ldots, x_i, \tau, s' \rangle).$$

Corollary 2 (Preservation of estimated performance)**.** *Let $\tilde{M}$ and $\tilde{M}^{2s}$ be the MLE-MDPs as above, constructed from $\mathcal{D}$ and $\mathcal{D}^{2s}$, respectively. Further, let $\tilde{\pi}$ be an arbitrary policy on $\tilde{M}$ and $\tilde{\pi}^{2s}$ the policy that extends π for $\tilde{M}^{2s}$ by choosing τ in all auxiliary states. Then $\rho(\tilde{\pi}, \tilde{M}) = \rho^{2s}(\tilde{\pi}^{2s}, \tilde{M}^{2s})$.*

We want to emphasize that while $\mathcal{D}^{2s}$ may contain more samples than $\mathcal{D}$, it does not yield any additional information. Rather, instead of viewing each transitions sample as an atomic data point, in $\mathcal{D}^{2s}$ transition samples are considered like a multi-step process. E.g, The sample $(s, a, s_3) \in \mathcal{D}$ would be transformed into the samples $\{(s, a, x_2), (x_2, \tau, x_3), (x_3, \tau, s_3)\} \in \mathcal{D}^{2s}$ which in the construction of the MLE-MDP are used to estimate the probabilities $P(s' \neq s_1 \mid s, a), P(s' \neq s_2 \mid s, a, s' \neq s_1)$ and $P(s' = s_3 \mid s, a, s' \neq s_1, s' \neq s_2)$, respectively. The probabilities of these events are mutually independent, but when multiplied give exactly $P(s_3 \mid s, a)$.

5 SPI in Two-Successor MDPs

In this section, we discuss how SPI can benefit from two-successor MDPs as constructed following our new transformation presented in Section 4. The dominating term in the bound N obtained by [Laroche *et al.*, 2019] is the branching factor of the MDP, which, without any prior information, has to necessarily be over-approximated by $|S|$ (cf. Section 3.2). We use our transformation above to bound the branching factor to $k = 2$, which allows us to provide stronger guarantees with the same data set (or conversely, require less data to guarantee a set maximum performance loss). Note that bounding the branching factor by any other constant can be achieved by a similar transformation as in Section 4, but $k = 2$ leads to an optimal bound [Wienhöft *et al.*, 2023].

Let $\tilde{M}$ and $\tilde{M}^{2s}$ be the MLE-MDPs inferred from data sets $\mathcal{D}$ and $\mathcal{D}^{2s}$, respectively. Further, let $\pi_\odot$ and $\pi_\odot^{2s}$ denote the optimal policies in these MLE-MDPs, constrained to the set of policies that follow π_b for state-action pairs $(s, a) \in \mathcal{U}$. Note that these optimal policies can easily be computed using, *e.g.*, standard value iteration. First, we show how to improve the admissible performance loss ζ in SPI on two-successor MDPs.

Lemma 1. *Let M^{2s} be a two-successor MDP with behavior policy π_b. Then, $\pi_\odot^{2s}$ is a ζ-approximately safe policy improvement over π_b with high probability $1 - \delta$, where:*

$$\zeta = \frac{4V_{max}}{1-\gamma}\sqrt{\frac{2}{N_\wedge}\log\frac{8|S||A|}{\delta}} + \tilde{\rho}^{2s},$$

with $\tilde{\rho}^{2s} = -\rho^{2s}(\pi_\odot^{2s}, \tilde{M}^{2s}) + \rho^{2s}(\pi_b, \tilde{M}^{2s})$.

For a general MDP M, we can utilize this result by first applying the transformation from Section 4.1.

Theorem 3 (Weissman-based tighter improvement guarantee)**.** *Let M be an MDP with behavior policy π_b. Then, $\pi_\odot$ is a ζ^{2s}-approximate safe policy improvement over π_b with high probability $1 - \delta$, where:*

$$\zeta^{2s} = \frac{4V_{max}}{1-\gamma}\sqrt{\frac{2}{N_\wedge^{2s}}\log\frac{8|S|^2|A|^2}{\delta}} - \rho(\pi_\odot, \tilde{M}) + \rho(\pi_b, \tilde{M}).$$

As for ζ^{SPIBB}, we can rearrange the equation to compute the number of necessary samples for a ζ^{2s}-safe improvement:

$$N_\wedge^{2s} = \frac{32V_{max}^2}{(\zeta^{2s})^2(1-\gamma)^2}\log\frac{8|S|^2|A|^2}{\delta}.$$

Note that ζ^{2s} and $N_\wedge^{2s}$ only depend on parameters of M and policy performances on $\tilde{M}$, which follows from Corollary 2 yielding $\rho(\pi_\odot, \tilde{M}) = \rho^{2s}(\pi_\odot, \tilde{M}^{2s})$. Hence, it is not necessary to explicitly compute the transformed MLE-MDP $\tilde{M}^{2s}$.

5.1 Uncertainty in Two-Successor MDPs

So far, the methods we outlined relied on a bound of the L_1-distance between a probability vector and its estimate based on a number of samples [Weissman *et al.*, 2003]. In this section, we outline a second method to tighten this bound for two-successor MDP and how to apply it to obtain a smaller approximation error ζ^β for a fixed $N_\wedge^\beta$.

Formally, given a 2sMDP M^{2s} and an error tolerance δ, we construct an error function $e\colon S \times A \to \mathbb{R}$ that ensures with probability $1 - \delta$ that $\|P(s,a) - \hat{P}(s,a)\|_1 \leq e(s,a)$ for all (s, a). To achieve this, we distribute δ uniformly over all states to obtain $\delta_T = \delta/|S|$, independently ensuring that for each state-action pair (s, a) the condition $\|P(s,a) - \hat{P}(s,a)\|_1 \leq e(s,a)$ is satisfied with probability at least $1 - \delta_T$.

We now fix a state-action pair (s, a). Since we are dealing with a two-successor MDP, there are only two successor states, s_1 and s_2. To bound the error function, we view each sample of action a in state s as a Bernoulli trial. As shorthand notation, we define $p = P(s, a, s_1)$, and consequently we have $1 - p = P(s, a, s_2)$. Using a uniform prior over p and given a data set $\mathcal{D}$ in which (s, a, s_1) occurs k_1 times and (s, a, s_2) occurs k_2 times, the posterior probability over p is given by a beta distribution with parameters $k_1{+}1$ and $k_2{+}1$, *i.e.*, $\Pr(p \mid \mathcal{D}) \sim B(k_1{+}1, k_2{+}1)$ [Jaynes, 2003]. We can express the error function in terms of the probability of p being contained in a given interval $[\underline{p}, \overline{p}]$ as $e(s, a) = \overline{p} - \underline{p}$.

The task that remains is to find such an interval $[\underline{p}, \overline{p}]$ for which we can guarantee with probability δ_T that p is contained within it. Formally, we can express this via the incomplete regularized beta function I, which in turn is defined as the cumulative density function of the beta distribution B:

$$\mathbb{P}(p \in [\underline{p}, \overline{p}]) = I_{\underline{p},\overline{p}}(k_1{+}1, k_2{+}1).$$

We show that we can define the smallest such interval in terms of the inverse incomplete beta function [Temme, 1992], denoted as I_δ^{-1}.

Lemma 2. *Let $k \sim Bin(n, p)$ be a random variable according to a binomial distribution. Then the smallest interval $[\underline{p}, \overline{p}]$ for which*

$$\mathbb{P}\left(p \in \left[\underline{p}, \overline{p}\right]\right) \geq 1 - \delta_T$$

holds, has size

$$\overline{p} - \underline{p} \leq 1 - 2I^{-1}_{\delta_T/2}\left(\frac{n}{2} + 1, \frac{n}{2} + 1\right).$$

Next, we show how to utilize this bound for the interval size in MDPs with arbitrary topology. The core idea is the same as in Theorem 3: We transform the MDP into a 2sMDP and apply the error bound $e(s, a) = \overline{p} - \underline{p}$ from Lemma 2.

Theorem 4 (Beta-based tighter improvement guarantee)**.** *Let M be an MDP with behavior policy π_b. Then, $\pi_\odot$ is a ζ^β-approximate safe policy improvement over π_b with high probability $1 - \delta$, where:*

$$\zeta^\beta = \frac{4V_{max}}{1-\gamma}\left(1 - I^{-1}_{\delta_T/2}\left(\frac{N^\beta_\wedge}{2} + 1, \frac{N^\beta_\wedge}{2} + 1\right)\right) + \tilde{\rho},$$

with $\delta_T = \frac{\delta}{|S|^2|A|^2}$, and $\tilde{\rho} = -\rho(\pi_\odot, \tilde{M}) + \rho(\pi_b, \tilde{M})$.

There is no closed formula to directly compute $N^\beta_\wedge$ for a given ζ^β. However, for a given admissible performance loss ζ, we can perform a binary search to obtain the smallest natural number $N^\beta_\wedge$ such that $\zeta^\beta \leq \zeta$ given in Theorem 4.

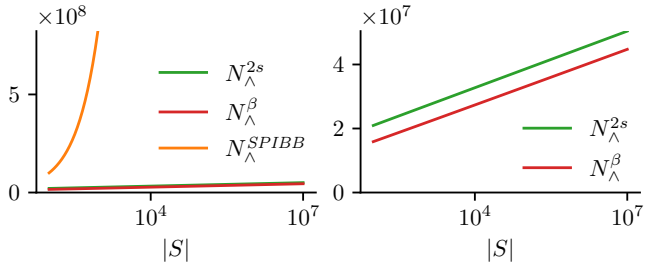
Comparison of Different $N_\wedge$. In the context of SPI, finding an $N_\wedge$ that is as small as possible while still guaranteeing ζ-approximate improvement is the main objective. An overview of the different ζ and $N_\wedge$ that are available is given in Table 1. Comparing the equations for different $N_\wedge$, we immediately see that $N^{\text{2s}}_\wedge \leq N^{\text{SPIBB}}_\wedge$ if and only if $2^{|S|} \geq 4|S||A|$. This means the only MDPs where standard SPIBB outperforms our 2sMDP approach are environments with a small state-space but a large action-space. By Lemma 2, we have that the error term $e(s, a)$ used to compute ζ^β is minimal in the 2sMDP[1], and in particular it is smaller than the error term used to compute ζ^{2s}. Thus we always have $N^\beta_\wedge \leq N^{\text{2s}}_\wedge$. In case $2^{|S|} < 4|S||A|$ it is also possible to compute both $N^{\text{SPIBB}}_\wedge$, and $N^\beta_\wedge$ and simply choose the smaller one.

6 Implementation and Evaluation

We provide an evaluation[2] of our approach from two different perspectives. First, a theoretical evaluation of how the different $N_\wedge$ depend on the size of a hypothetical MDP, and second, a practical evaluation to investigate how smaller $N_\wedge$ values translate to the performance of the improved policies.

[1]Technically, Lemma 2 allows for arbitrary parameters while the SPIBB algorithm only allows integers for the number of samples, and thus integer parameters in the inverse beta function, so ζ^β is only minimal for even $N^\beta_\wedge$. However, we can easily adapt the equation for odd $N^\beta_\wedge$ by replacing $N^\beta_\wedge$ by $N^\beta_\wedge - 1$ and $N^\beta_\wedge + 1$, respectively.

[2]Code available at https://github.com/LAVA-LAB/improved_spi.



(a) $N^{\text{SPIBB}}_\wedge$, $N^{\text{2s}}_\wedge$ and $N^\beta_\wedge$. (b) $N^{\text{2s}}_\wedge$ and $N^\beta_\wedge$.

Figure 3: Required number of samples for different $|S|$ with $|A| = 4, V_{max} = 1, \gamma = 0.95, \delta = 0.1$ and $\zeta = 0.1$.

6.1 Example Comparison of Different $N_\wedge$

To render the theoretical differences between the possible $N_\wedge$ discussed at the end of Section 5 more tangible, we now give a concrete example.

We assume a hypothetical MDP with $|A| = 4$, $V_{max} = 1$, $\gamma = 0.95$, and SPIBB parameters $\delta = 0.1$ and $\zeta = 0.1$. For varying sizes of the state-space, we compute all three sample size constraints: $N^{\text{SPIBB}}_\wedge$, $N^{\text{2s}}_\wedge$, and $N^\beta_\wedge$. The results are shown in Figure 3, where Figure 3a shows the full plot and Figure 3b provides an excerpt to differentiate between the $N^{\text{2s}}_\wedge$ and $N^\beta_\wedge$ plots by scaling down the y-axis. Note that the x-axis, the number of states in our hypothetical MDP, is on a log-scale. We see that $N^{\text{SPIBB}}_\wedge$ grows linearly with the number of states, whereas $N^{\text{2s}}_\wedge$ and $N^\beta_\wedge$ are logarithmic in the number of states. Further, we note that $N^\beta_\wedge$ is significantly below $N^{\text{2s}}_\wedge$, which follows from Lemma 1. Finally, the difference between $N^{\text{SPIBB}}_\wedge$ and $N^{\text{2s}}_\wedge$ is for small MDPs of around a hundred states already a factor 10.

Discussion. While we show that a significant reduction of the required number of samples per state-action pair $N_\wedge$ is possible via our two approaches, we note that even for small MDPs (*e.g.*, $|S| = 100$) we still need over 10 million samples per state-action pair to guarantee that an improved policy is safe *w.r.t.* the behavior policy. That is, with probability $1 - \delta = 0.9$, an improved policy will have an admissible performance loss of at most $\zeta = 0.1$, which is infeasible in practice. Nevertheless, a practical evaluation of our approaches is possible taking on a different perspective, which we address in the next section.

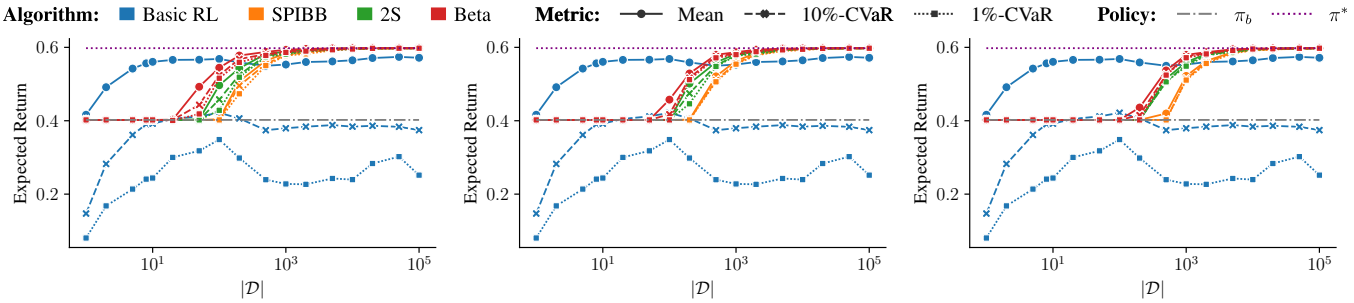
6.2 Evaluation in SPIBB

We integrate our novel results for computing ζ^{2s}, ζ^β, $N^{\text{2s}}_\wedge$, and $N^\beta_\wedge$ into the implementation of SPIBB [Laroche *et al.*, 2019].

Benchmarks. We consider two standard benchmarks used in SPI and one other well-known MDP: the 25-state *Grid-world* proposed by Laroche *et al.* [2019], the 25-state *Wet Chicken* benchmark [Hans and Udluft, 2009], which was used to evaluate SPI approaches by Scholl *et al.* [2022], and a 376-state instance of *Resource Gathering* proposed by Barrett and Narayanan [2008].

Method	Admissible performance loss ζ	Number of samples $N_\wedge$
Standard SPI [Petrik *et al.*, 2016]	$\zeta^{\text{SPI}} = \frac{2\gamma V_{max}}{1-\gamma}\sqrt{\frac{2}{N_\wedge^{\text{SPI}}}\log\frac{2\lvert S\rvert\lvert A\rvert 2^{\lvert S\rvert}}{\delta}}$	$N_\wedge^{\text{SPI}} = \frac{8V_{max}^2}{{\zeta^{\text{SPI}}}^2(1-\gamma)^2}\log\frac{2\lvert S\rvert\lvert A\rvert 2^{\lvert S\rvert}}{\delta}$ $(\star)$
Standard SPIBB [Laroche *et al.*, 2019]	$\zeta^{\text{SPIBB}} = \frac{4V_{max}}{1-\gamma}\sqrt{\frac{2}{N_\wedge^{\text{SPIBB}}}\log\frac{2\lvert S\rvert\lvert A\rvert 2^{\lvert S\rvert}}{\delta}} + \tilde{\rho}$	$N_\wedge^{\text{SPIBB}} = \frac{32V_{max}^2}{{\zeta^{\text{SPIBB}}}^2(1-\gamma)^2}\log\frac{2\lvert S\rvert\lvert A\rvert 2^{\lvert S\rvert}}{\delta}$
Two-Successor SPIBB (Theorem 3)	$\zeta^{\text{2s}} = \frac{4V_{max}}{1-\gamma}\sqrt{\frac{2}{N_\wedge^{\text{2s}}}\log\frac{8\lvert S\rvert^2\lvert A\rvert^2}{\delta}} + \tilde{\rho}$	$N_\wedge^{\text{2s}} = \frac{32V_{max}^2}{(\zeta^{\text{2s}})^2(1-\gamma)^2}\log\frac{8\lvert S\rvert^2\lvert A\rvert^2}{\delta}$
Inverse beta SPIBB (Theorem 4)	$\zeta^{\beta} = \frac{4V_{max}}{1-\gamma}\left(1 - 2I^{-1}_{\delta_T/2}\left(\frac{N_\wedge^\beta}{2}+1, \frac{N_\wedge^\beta}{2}+1\right)\right) + \tilde{\rho}$	No closed formula available (use binary search to compute)

Table 1: Overview of the different ζ and $N_\wedge$ we obtain, where $\delta_T = \frac{\delta}{\lvert S\rvert^2\lvert A\rvert^2}$ and $\tilde{\rho} = -\rho(\pi_\odot, \tilde{M}) + \rho(\pi_b, \tilde{M})$ is the difference in performance between optimal and behavior policy on the MLE-MDP. $(\star)$ Standard SPI requires at least $N_\wedge^{\text{SPI}}$ samples in *all* state-action pairs.



(a) $N_\wedge^{\text{SPIBB}} = 100, N_\wedge^{\text{2s}} = 55, N_\wedge^{\beta} = 27.$ (b) $N_\wedge^{\text{SPIBB}} = 200, N_\wedge^{\text{2s}} = 110, N_\wedge^{\beta} = 67.$ (c) $N_\wedge^{\text{SPIBB}} = 400, N_\wedge^{\text{2s}} = 219, N_\wedge^{\beta} = 146.$

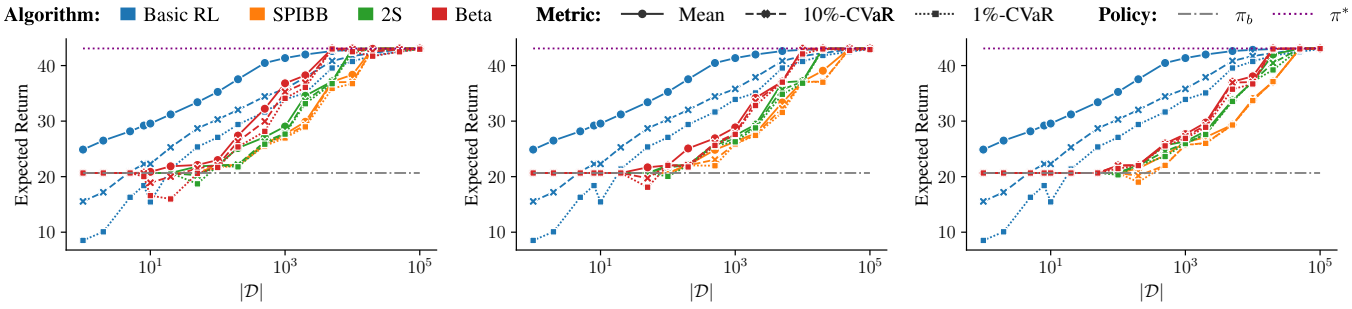
Figure 4: Safe policy improvement on the Gridworld environment.

Behavior policy. For the Gridworld, we use the same behavior policy as [Laroche *et al.*, 2019]. For the Wet Chicken environment, we use Q-Learning with a softmax function to derive a behavior policy. The behavior policy of Resource Gathering was derived from the optimal policy by selecting each non-optimal action with a probability of `1e-5`.

Methodology. Recall that in the standard SPIBB approach, $N_\wedge$ is used as a hyperparameter, since the actual $N_\wedge$ for reasonable δ and ζ are infeasible. While our methods improve significantly on $N_\wedge$, the values we obtain are still infeasible in practice, as discussed in Section 6.1. We still use $N_\wedge^{\text{SPIBB}}$ as a hyperparameter, and then run the SPIBB algorithm and compute the resulting ζ^{SPIBB}. This ζ^{SPIBB} is consequently used to compute the values $N_\wedge^{\text{2s}}$ and $N_\wedge^{\beta}$ that ensure the same performance loss. We then run SPIBB again with these two values for $N_\wedge$. As seen in the previous experiment, and detailed at the end of Section 5, for most MDPs – including our examples – we have $N_\wedge^{\beta} \leq N_\wedge^{\text{2s}} \leq N_\wedge^{\text{SPIBB}}$ for a fixed ζ.

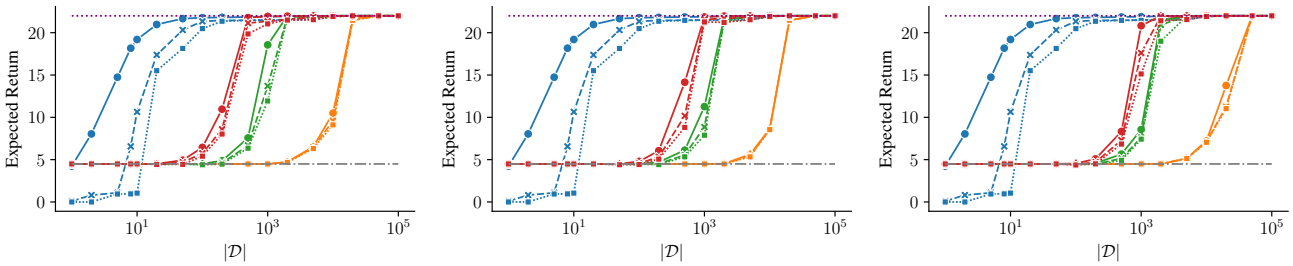
Evaluation metrics. For each data set size, we repeat each experiment 1000 times and report the mean performance of the learned policy, as well as the 10% and 1% conditional value at risk (CVaR) values [Rockafellar and Uryasev, 2000], indicating the mean performance of the worst 10% and 1% runs. To give a complete picture, we also include the performance of basic RL (dynamic programming on the MLE-MDP [Sutton and Barto, 1998]), the behavior policy π_b, and the optimal policy π^* of the underlying MDP.

Results. We present the results for the Gridworld, Wet Chicken, and Resource Gathering environments for three different hyperparameters $N_\wedge^{\text{SPIBB}}$ in Figures 4, 5, and 6, respectively. In all instances, we see similar and improved behaviors as we presumed by sharpening the sampling bounds with our new approaches. Smaller values for $N_\wedge$ typically require smaller data sets for a policy to start improving, and this is precisely what our methods set out to do. In particular, we note that our methods (2S and Beta) are quicker to converge to an optimal policy than standard SPIBB. Beta is, as expected, the fastest, and starts to improve over the behavior policy for data sets about half the size compared to SPIBB in the Gridworld. Further, while theoretically, the factor between the different $N_\wedge$ does not directly translate to the whole data set size, we see that in practice on all three benchmarks this is roughly the case. Finally, we note that Basic RL is unreliable compared to the SPI methods, as seen by the CVaR values being significantly below the baseline performance for several data set sizes in all three environments. This is as expected and in accordance with well-established results.



(a) $N_\wedge^{\text{SPIBB}} = 60, N_\wedge^{\text{2s}} = 34, N_\wedge^{\beta} = 10.$ (b) $N_\wedge^{\text{SPIBB}} = 120, N_\wedge^{\text{2s}} = 67, N_\wedge^{\beta} = 36.$ (c) $N_\wedge^{\text{SPIBB}} = 180, N_\wedge^{\text{2s}} = 101, N_\wedge^{\beta} = 61.$

Figure 5: Safe policy improvement on the Wet Chicken environment.



(a) $N_\wedge^{\text{SPIBB}} = 600, N_\wedge^{\text{2s}} = 43, N_\wedge^{\beta} = 12.$ (b) $N_\wedge^{\text{SPIBB}} = 800, N_\wedge^{\text{2s}} = 57, N_\wedge^{\beta} = 25.$ (c) $N_\wedge^{\text{SPIBB}} = 1000, N_\wedge^{\text{2s}} = 71, N_\wedge^{\beta} = 37.$

Figure 6: Safe policy improvement on the Resource Gathering environment.

7 Related Work

A variant of our transformation from MDP to 2sMDP was introduced by Mayr and Munday [2023], utilizing binary trees built from auxiliary states as gadgets. Similar to our construction, Junges *et al.* [2018] transform a partially observable MDP (POMDP) [Kaelbling *et al.*, 1998; Spaan, 2012] into a *simple POMDP*, where each state has either one action choice, and an arbitrary number of successor states, or where there are multiple actions available but each action has a single successor state. The same transformation was applied to *uncertain* POMDPs [Cubuktepe *et al.*, 2021].

Besides the main approaches to SPI mentioned in Section 3, there are a number of other noteworthy works in this area. SPIBB has been extended to *soft baseline bootstrapping* in [Nadjahi *et al.*, 2019], where instead of either following the behavior policy or the optimal policy in the MLE-MDP in a state-action pair, randomization between the two is applied. However, the theoretical guarantees of this approach rely on an assumption that rarely holds [Scholl *et al.*, 2022].

Incorporating structural knowledge of the environment has been shown to improve the sample complexity of SPI algorithms [Simão and Spaan, 2019a; Simão and Spaan, 2019b]. It is also possible to deploy the SPIBB algorithm in problems with large state space using MCTS [Castellini *et al.*, 2023]. For a more detailed overview of SPI approaches and an empirical comparison between them, see [Scholl *et al.*, 2022]. For an overview of how these algorithms scale in the number of states, we refer to [Brandfonbrener *et al.*, 2022].

Other related work investigated how to relax some of the assumptions SPI methods make. In [Simão *et al.*, 2020], a method for estimating the behavior policy is introduced, relaxing the need to know this policy. Finally, a number of recent works extend the scope and relax common assumptions by introducing SPI in problems with partial observability [Simão *et al.*, 2023], non-stationary dynamics [Chandak *et al.*, 2020], and multiple objectives [Satija *et al.*, 2021].

Finally, we note that SPI is a specific *offline* RL problem [Levine *et al.*, 2020], which has seen significant advances recently [Kidambi *et al.*, 2020; Yu *et al.*, 2020; Kumar *et al.*, 2020; Smit *et al.*, 2021; Yu *et al.*, 2021; Rigter *et al.*, 2022]. While these approaches may be applicable to high dimensional problems such as control tasks and problems with large observation space [Fu *et al.*, 2020], they often ignore the reliability aspect of improving over a baseline policy, as SPI algorithms do. Nevertheless, it remains a challenge to bring SPI to high-dimensional problems.

8 Conclusion

We presented a new approach to safe policy improvement that reduces the required size of data sets significantly. We derived new performance guarantees and applied them to state-of-the-art approaches such as SPIBB. Specifically, we introduced a novel transformation to the underlying MDP model that limits the branching factor, and provided two new ways of computing the admissible performance loss ζ and the sample size constraint $N_\wedge$, both exploiting the limited branching factor in SPI(BB). This improves the overall performance of SPI algorithms, leading to more efficient use of a given data set.

Contribution Statement

Patrick Wienhöft and Marnix Suilen share first authorship, contributing equally to the development and implementation of the method. The other authors contributed with discussions, ideas, and the presentation of the method.

Acknowledgments

The authors were partially supported by the DFG through the Cluster of Excellence EXC 2050/1 (CeTI, project ID 390696704, as part of Germany's Excellence Strategy), the TRR 248 (see https://perspicuous-computing.science, project ID 389792660), the NWO grants OCENW.KLEIN.187 (Provably Correct Policies for Uncertain Partially Observable Markov Decision Processes) and NWA.1160.18.238 (PrimaVera), and the ERC Starting Grant 101077178 (DEUCE).

References

[Barrett and Narayanan, 2008] Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *ICML*, pages 41–47. ACM, 2008.

[Brandfonbrener *et al.*, 2022] David Brandfonbrener, Remi Tachet des Combes, and Romain Laroche. Incorporating explicit uncertainty estimates into deep offline reinforcement learning. *arXiv preprint arXiv:2206.01085*, 2022.

[Castellini *et al.*, 2023] Alberto Castellini, Federico Bianchi, Edoardo Zorzi, Thiago D. Simão, Alessandro Farinelli, and Matthijs T. J. Spaan. Scalable Safe Policy Improvement via Monte Carlo Tree Search. In *ICML*, 2023.

[Chandak *et al.*, 2020] Yash Chandak, Scott M. Jordan, Georgios Theocharous, Martha White, and Philip S. Thomas. Towards safe policy improvement for non-stationary MDPs. In *NeurIPS*, pages 9156–9168. Curran Associates, Inc., 2020.

[Cubuktepe *et al.*, 2021] Murat Cubuktepe, Nils Jansen, Sebastian Junges, Ahmadreza Marandi, Marnix Suilen, and Ufuk Topcu. Robust finite-state controllers for uncertain POMDPs. In *AAAI*, pages 11792–11800. AAAI Press, 2021.

[Fu *et al.*, 2020] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[Hans and Udluft, 2009] Alexander Hans and Steffen Udluft. Efficient uncertainty propagation for reinforcement learning with limited data. In *ICANN (1)*, pages 70–79. Springer, 2009.

[Jaynes, 2003] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[Junges *et al.*, 2018] Sebastian Junges, Nils Jansen, Ralf Wimmer, Tim Quatmann, Leonore Winterer, Joost-Pieter Katoen, and Bernd Becker. Finite-state controllers of POMDPs using parameter synthesis. In *UAI*, pages 519–529. AUAI Press, 2018.

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

[Kidambi *et al.*, 2020] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *NeurIPS*, pages 21810–21823. Curran Associates, Inc., 2020.

[Kumar *et al.*, 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *NeurIPS*, pages 1179–1191. Curran Associates, Inc., 2020.

[Lange *et al.*, 2012] Sascha Lange, Thomas Gabel, and Martin A. Riedmiller. Batch reinforcement learning. In *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 45–73. Springer, 2012.

[Laroche *et al.*, 2019] Romain Laroche, Paul Trichelair, and Remi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *ICML*, pages 3652–3661. PMLR, 2019.

[Levine *et al.*, 2020] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[Mayr and Munday, 2023] Richard Mayr and Eric Munday. Strategy Complexity of Point Payoff, Mean Payoff and Total Payoff Objectives in Countable MDPs. *Logical Methods in Computer Science*, Volume 19, Issue 1, March 2023.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.

[Nadjahi *et al.*, 2019] Kimia Nadjahi, Romain Laroche, and Rémi Tachet des Combes. Safe policy improvement with soft baseline bootstrapping. In *ECML/PKDD (3)*, pages 53–68. Springer, 2019.

[Petrik *et al.*, 2016] Marek Petrik, Mohammad Ghavamzadeh, and Yinlam Chow. Safe policy improvement by minimizing robust baseline regret. In *NIPS*, pages 2298–2306. Curran Associates, Inc., 2016.

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

[Ramakrishnan *et al.*, 2020] Ramya Ramakrishnan, Ece Kamar, Debadeepta Dey, Eric Horvitz, and Julie Shah. Blind spot detection for safe sim-to-real transfer. *J. Artif. Intell. Res.*, 67:191–234, 2020.

[Rigter *et al.*, 2022] Marc Rigter, Bruno Lacerda, and Nick Hawes. RAMBO-RL: Robust adversarial model-based offline reinforcement learning. In *NeurIPS*, pages 16082–16097. Curran Associates, Inc., 2022.

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

[Satija *et al.*, 2021] Harsh Satija, Philip S. Thomas, Joelle Pineau, and Romain Laroche. Multi-objective SPIBB: seldonian offline policy improvement with safety constraints in finite MDPs. In *NeurIPS*, pages 2004–2017, 2021.

[Scholl *et al.*, 2022] Philipp Scholl, Felix Dietrich, Clemens Otte, and Steffen Udluft. Safe policy improvement approaches on discrete Markov decision processes. In *ICAART (2)*, pages 142–151. SCITEPRESS, 2022.

[Silver *et al.*, 2018] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

[Simão and Spaan, 2019a] Thiago D. Simão and Matthijs T. J. Spaan. Safe policy improvement with baseline bootstrapping in factored environments. In *AAAI*, pages 4967–4974. AAAI Press, 2019.

[Simão and Spaan, 2019b] Thiago D. Simão and Matthijs T. J. Spaan. Structure learning for safe policy improvement. In *IJCAI*, pages 3453–3459. ijcai.org, 2019.

[Simão *et al.*, 2020] Thiago D. Simão, Romain Laroche, and Rémi Tachet des Combes. Safe policy improvement with an estimated baseline policy. In *AAMAS*, pages 1269–1277. IFAAMAS, 2020.

[Simão *et al.*, 2023] Thiago D. Simão, Marnix Suilen, and Nils Jansen. Safe policy improvement for POMDPs via finite-state controllers. In *AAAI*. AAAI Press, 2023.

[Smit *et al.*, 2021] Jordi Smit, Canmanie Ponnambalam, Matthijs T.J. Spaan, and Frans A. Oliehoek. PEBL: Pessimistic ensembles for offline deep reinforcement learning. In *IJCAI Workshop on Robust and Reliable Autonomy in the Wild (R2AW)*, 2021.

[Spaan, 2012] Matthijs T. J. Spaan. Partially observable Markov decision processes. In *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 387–414. Springer, 2012.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

[Temme, 1992] N.M. Temme. Asymptotic inversion of the incomplete beta function. *Journal of Computational and Applied Mathematics*, 41(1):145–157, 1992.

[Thomas *et al.*, 2015] Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *ICML*, pages 2380–2388. PMLR, 2015.

[Weissman *et al.*, 2003] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, and Marcelo J. Weinberger. Inequalities for the L1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

[Wienhöft *et al.*, 2023] Patrick Wienhöft, Marnix Suilen, Thiago D. Simão, Clemens Dubslaff, Christel Baier, and Nils Jansen. More for less: Safe policy improvement with stronger performance guarantees. *arXiv preprint arXiv:2305.07958*, 2023.

[Yu *et al.*, 2020] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *NeuRIPS*, pages 14129–14142. Curran Associates, Inc., 2020.

[Yu *et al.*, 2021] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: conservative offline model-based policy optimization. In *NeurIPS*, pages 28954–28967. Curran Associates, Inc., 2021.

[Zhao *et al.*, 2020] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *SSCI*, pages 737–744. IEEE, 2020.