2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Safe Policy Improvement with Baseline Bootstrapping under State Abstraction

Yuan Zhuang

Nanjing University

Abstract. This paper studies the Safety Policy Improvement (SPI) problem in Batch Reinforcement Learning (Batch RL), which aims to train a policy from a fixed dataset without environment interaction, while ensuring its performance is no worse than the behavior policy used for data collection. Most existing methods often require a substantial amount of historical data to ensure sufficient confidence in the performance of the learned policy. However, the fixed dataset is often limited, which causes the learning overly conservative. To address this issue, we investigate the integration of state abstraction into the SPIBB framework to improve sample efficiency. While state abstraction has been widely used to improve sample efficiency, it traditionally lacks mechanisms for providing performance guarantees. We bridge this gap by deriving theoretical performance guarantees of policies learned from SPIBB under state abstraction. Empirical results show that our method achieves comparable or better policy improvement using fewer samples than the original SPIBB algorithm.

Keywords: Batch Reinforcement Learning, Safe policy Improvement with Baseline Bootstrapping, State Abstraction.

1 Introduction

Reinforcement Learning (RL) has achieved remarkable progress in long-term planning, global optimization, and sequential decision-making [1]. Representative advances include Deep Q-Networks (DQNs) for Atari games, AlphaGo's victory over the Go world champion [2] and RL from Human Feedback (RLHF), a key technique behind aligning large language models with human intent [3]. Traditional RL relies on trial-and-error interaction with the environment to learn optimal policies. However, this paradigm is often infeasible in safety-critical domains such as healthcare, industrial control, and finance, where exploration can be costly or risky. Batch RL addresses this limitation by enabling policy learning from fixed datasets without requiring further interaction with the environment [4, 5].

Safe Policy Improvement (SPI) is a fundamental topic in Batch RL, focusing on learning a policy from fixed data that performs at least as well as the *behavior policy* that generated it [6-9]. SPI also holds significant practical value. For example, policies

may need to be deployed simultaneously across many independent devices (e.g., widespread software updates on smartphones), where failures can lead to extremely high repair costs. Moreover, policy evaluation may require a long period of time (e.g., in crop management or clinical trials), during which deploying a bad policy could cause severe consequences. Research on SPI can greatly reduce the risks associated with such scenarios, ensuring the stability and consistency of deployed policies.

Most SPI approaches focus on the model-based RL paradigm to address this fundamental problem in the context of infinite-horizon discounted Markov decision processes (MDPs) [6-9]. Safe Policy Improvement with Baseline Bootstrapping (SPIBB) is a representative SPI method that has inspired various extensions, including adaptations to different MDP variants [9, 10] and improved compatibility with diverse behavior policies [11, 12]. It that provides confidence bounds on the improvement of the learned policy over the behavior policy. For instance, SPIBB ensures with 0.99 probability that the learned policy outperforms the behavior policy within a performance margin of 0.1. Such performance margin serves as a confidence bounds, that is essential for practitioners when deciding whether to deploy a new policy in true environments. A smaller confidence bound implies that the new policy is more likely to outperform the behavior policy.

SPIBB approach impose constraints that restrict training, allowing policy learning only when the constraints are met. Typically, the constraints depend on the available data. As sufficient data is needed to accurately capture the agent-environment interaction dynamics. As a result, the learned optimal policy from these samples is more likely to perform well in the true environment, thus outperforming the behavior policy. However, since the available data is usually limited, SPIBB method often result in conservative training. As a result, improving sample efficiency has become a critical challenge for enabling the practical application of SPIBB in the true environment.

At the same time, state abstraction can reduce the size of an MDP by grouping similar states and has been applied in model-based reinforcement learning to lower learning complexity and enhance sample efficiency [13-15]. In this work, we adopt approximate stochastic bisimulation as the abstraction method, which merges states with similar transition dynamics and rewards into abstract states. This approach is naturally suited to improving the sample efficiency of the SPIBB algorithm, as it allows the sharing of samples across grouped states. Such sharing alleviates the issue of data scarcity and helps better satisfy the algorithm's constraints. However, since the similar states are not perfectly equivalent, the policies learned from these samples often differ from those learned without sharing. Consequently, it is necessary to derive new confidence bounds for the performance improvement of the policy learned via SPIBB under the framework of state abstraction.

In this work, we propose a SPIBB algorithm incorporating the state abstraction technique to improve sample efficiency in the batch reinforcement learning setting. Our main contribution is a theoretical confidence bound that quantifies the performance guarantee of the learned policy by SPIBB algorithm incorporating the state abstraction. We further demonstrate through empirical analysis that the proposed method can achieve greater policy improvement using significantly fewer samples than the original

SPIBB algorithm. These findings suggest that abstraction can be a powerful tool for safe and efficient policy learning in data-limited settings.

2 Preliminaries

2.1 MDPs and Reinforcement Learning

We briefly introduce the notations for Markov Decision Processes (MDPs) and Reinforcement Learning (RL). For a comprehensive introduction, we refer readers to the relevant literature [1, 16].

An MDP is defined by $M = (S, A, T, R, s_{init}, \gamma)$, where S is the state space, A is the action space, $R: S \times A \to \mathbb{R}$ is the reward function, where $R(s, a)$ represents the reward the agent receives after taking action a in state s, s_{init} is the initial state, and $\gamma \in [0,1)$ is the discount factor. The true environment is modelled as an unknown finite MDP $M^* = (S, A, R, T^*, \gamma, s_{init})$ with unknown transition probability T^*.

A policy is defined as $\pi: S \to \Delta(A)$, where $\Delta(A)$ denotes a probability distribution over the action set A. The value function of a policy π in MDP M is defined as $V_M^\pi(s) = E_{\pi,M}[\sum_{t\geq 0} \gamma^t R(s_t, a_t) | s_0 = s, a_t \sim \pi(s_t)]$, representing the expected discounted return when starting from state s and following π. The value of M is denoted as $\rho(\pi, M) = V_M^\pi(s_{init})$. The optimal policy over all policies $\Pi: \{\pi: S \to \Delta(A)\}$ is $\pi^* = \arg\max_{\pi\in\Pi} \rho(\pi, M)$, while the Π'-optimal policy over a subset $\Pi' \in \Pi$ is $\pi^*_{\Pi'} = \arg\max_{\pi\in\Pi'} \rho(\pi, M)$. The value function is upper bounded by $V_{max} \leq \frac{R_{max}}{1-\gamma}$, where R_{max} is the maximum reward.

In this paper, we consider the batch RL setting [4], where the algorithm does its best at learning a policy from a fixed set of experience. Given a dataset of transitions $D = \{(s_j, a_j, r_j, s'_j) | j \in [1, N]\}$, we denote by $N_D(s, a)$ the state-action pair counts, and by $N_D(s, a, s')$ the number of transitions from (s, a) to s'. A vanilla batch RL approach, referred to as Basic RL, adopts a model-based manner [17] by explicitly constructing a Maximum Likelihood Estimation (MLE) MDP $\widehat{M} = (S, A, R, \hat{T}, s_{init}, \gamma)$, where the estimated transition probability is given by:

$$\forall s, s' \in S, a \in A, \hat{T}(s'|s, a) = \frac{N_D(s, a, s')}{N_D(s, a)} \quad (1)$$

Once the model $\widehat{M}$ is constructed, the optimal policy can be derived through dynamic programming on $\widehat{M}$ [18], Q-learning with experience replay until convergence [19], etc.

If the estimated MDP $\widehat{M}$ closely approximates M^*, the optimal policy learned from $\widehat{M}$ may perform optimal in the true environment. However, datasets are often limited, particularly in high-risk fields such as healthcare and finance. With insufficient data, the learned policy may lack robustness in state-action pairs with fewer samples, potentially leading to high-risk decisions.

2.2 Safe Policy Improvement

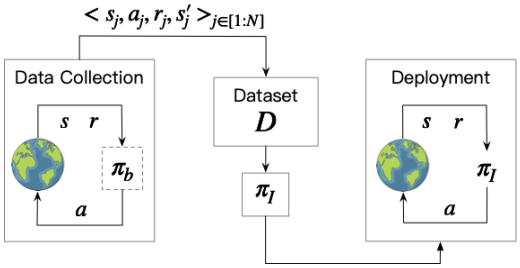

Fig.1. Illustration of the SPI problem in Batch RL

This section reviews the SPI problem and a representative state-of-the-art solution, which will subsequently be extended to incorporate state abstraction.

The Safe Policy Improvement (SPI) problem focuses on guaranteeing the performance of policies learned from fixed datasets in the true environment [6-9]. Fig. 1 illustrates the framework of the SPI problem. This problem typically assumes a behavior policy π_b, which generates the fixed dataset $D = \{(s_j, a_j, r_j, s'_j)|j \in [1, N]\}$. The goal of SPI is to learn a policy π_I from D such that, with probability at least $1 - \delta$, its performance deviates from that of the behavior policy π_b by no more than an admissible performance loss ζ:

$$\rho(\pi_I, M^*) \geq \rho(\pi_b, M^*) - \zeta \tag{2}$$

Percentile criterion [6, 7]**.** Given hyperparameters δ and ζ, the SPI problem can be formalized as a percentile criterion optimization problem. As a first step, constructing a set of admissible MDPs provides a robust surrogate for the unknown true environment M^*. Formally, the admissible MDPs set is defined as:

$$\begin{aligned} \Xi_e^{\widehat{M}} = \{M = (S, A, R, T, \gamma, s_{init}) | \ \forall (s, a) \in S \times A, s' \in S \\ s.t. \left\| T(s'|s, a) - \hat{T}(s'|s, a) \right\|_1 \leq e(s, a)\} \end{aligned} \tag{3}$$

Here, $e(s, a)$ denotes an error function, which captures the maximum L_1 distance between transition functions across all state-action pairs in the MDP. The L_1 distance is defined as the sum of absolute differences between the corresponding components of two probability distributions.

Secondly, an error function e is introduced such that the uncertainty set $\Xi_e^{\widehat{M}}$ contains the true MDP M^* with high probability at least $1 - \delta$. This enables the SPI problem to be formulated as a percentile criterion optimization problem, where the objective is to learn an improved policy π_I based on the estimated model $\widehat{M}$, such that π_I approximately outperforms the behavior policy π_b across all MDPs in $\Xi_e^{\widehat{M}}$. The formal definition is as follows:

$$\pi_I = \arg\max_{\pi} \rho\left(\pi, \widehat{M}\right), \text{s.t.} \ \forall M \in \Xi_e^{\widehat{M}}, \rho(\pi, M) \geq \rho(\pi_b, M) - \zeta \tag{4}$$

Since $\Xi_e^{\widehat{M}}$ contains the true environment M^* with probability at least $1-\delta$, any policy that satisfies this condition is guaranteed, with probability $1-\delta$, to outperform the behavior policy in the true environment.

Finally, a safety constraint is derived from the percentile criterion optimization problem. This constraint specifies a threshold $N_\wedge$ on the number of samples $N_D(s,a)$ required for each state-action pair $(s,a) \in S \times A$, defined as follows:

$$\forall (s,a) \in S \times A, N_D(s,a) \geq N_\wedge = \frac{8V_{max}^2}{\zeta^2(1-\gamma)^2} \log \frac{2|S||A|2^{|S|}}{\delta} \tag{5}$$

This constraint restricts the policy learning is permitted only when the sample size exceeds this threshold. This ensure that the resulting policy is a feasible solution to the optimization problem, thereby meeting the requirements of safe policy improvement.

Safe Policy Improvement with Baseline Bootstrapping (SPIBB) [7]. The bound in Equation (5) must hold for all state-action pairs, which may not be satisfied by the dataset D, thereby limiting policy learning. To address this limitation, the SPIBB algorithm relaxes this requirement by permitting the constraint in Equation (5) to be violated for certain state-action pairs. The set B is defined to include all state-action pairs whose visitation counts fall below the threshold $N_\wedge$:

$$B = \{(s,a) \in S \times A | N_D(s,a) \leq N_\wedge\} \tag{6}$$

The SPIBB algorithm computes an improved policy π_I on the estimated MDP $\widehat{M}$, similarly to standard policy optimization, but with an additional constraint: for all state-action pairs $\forall (s,a) \in B$, the improved policy must match the behavior policy, i.e., $\pi_I(a|s) = \pi_b(a|s)$. Under this constraint, π_I constitutes a ζ-approximately safe improvement over the behavior policy π_b with probability $1-\delta$. Based on theoretical analysis, an admissible performance loss ζ is derived and defined as follows:

$$\zeta = \frac{4V_{max}}{1-\gamma} \sqrt{\frac{2}{N_\wedge} \log \frac{2|S||A|2^{|S|}}{\delta}} - \rho\left(\pi_I, \widehat{M}\right) + \rho\left(\pi_b, \widehat{M}\right) \tag{7}$$

SPIBB allows users to set the threshold $N_\wedge$ and enables policy learning even when some state-action pairs have fewer samples than this threshold, thereby improving efficiency. Its core is the derivation of the admissible performance loss ζ, which bounds the performance gap within which the improved policy π_I is guaranteed to outperform the behavior policy π_b. A smaller ζ implies greater confidence in π_I, supporting its safe replacement of π_b in the true environment.

SPIBB is an important method for learning policies with performance guarantees while maintaining effective learning. It has inspired numerous extensions, including adaptations to various MDP variants and improved compatibility with a wide range of behavior policies.

2.3 RL with State Abstraction

State abstraction technique could maps the original states space S in an MDP into smaller abstract state $\bar{S}$ in an abstract MDP, which reduces the problem complexity while maintaining a bounded loss with respect to the original problem [20, 21]. Recently, fueled by rapid advancements in reinforcement learning, approximate stochastic bisimulation, an approximate state abstraction technique, has been integrated into the RL paradigm to enhance sample efficiency [13, 15, 22, 23]. In their setting, there exits an approximate stochastic bisimulation function $\phi: S \to \bar{S}$. And the agent acts in an MDP that returns states s, but instead of observing the true state s, the agent observes abstract states $\phi(s)$.

In this section, we introduce the notion of approximate stochastic bisimulation and discuss how to learn policies in RL based on approximate stochastic bisimulation.

Definition 1 (approximate stochastic bisimulation, ϕ) [20, 21]. Given two states $s_1, s_2 \in S$, if for any action $a \in A$, the difference in their transition probabilities to any abstract state $\bar{s}' \in \bar{S}$ is bounded by η, then s_1 and s_2 can be mapped into the same abstract state under the function ϕ, i.e.:

$$\phi(s_1) = \phi(s_2) \Longrightarrow \forall \bar{s}' \in \bar{S}, a \in A: |T(\bar{s}'|s_1, a) - T(\bar{s}'|s_2, a)| \leq \eta \tag{8}$$

Under the approximate stochastic bisimulation function $\phi: S \to \bar{S}$, the original dataset $D = \{(s_j, a_j, r_j, s_j')|j \in [1, N]\}$ collected from the underlying MDP can be transformed into an abstracted dataset $\mathcal{D} = \{(\bar{s}_j, a_j, r_j, \bar{s}_j')|j \in [1, N]\}$, where $\bar{s}_j = \phi(s_j)$ and $\bar{s}_j' = \phi(s_j, a_j, r_j, s_j')$. This dataset captures the transitions between abstract states, thereby enabling policy learning in the abstracted state space.

An abstract policy $\bar{\pi}: \bar{S} \to \Delta(A)$ can be optimized using batch RL algorithm in a model-based manner. Specifically, we consider the estimated abstract MDP $\hat{\bar{M}} = \left(\bar{S}, A, \hat{\bar{T}}, R, \bar{s}_0, \gamma\right)$, where $\hat{\bar{T}}$ denotes the transition dynamics over the abstract states, which can be derived from the abstracted dataset $\mathcal{D}$ as follows:

$$\forall \bar{s}, \bar{s}' \in S, a \in A, \hat{\bar{T}}(\bar{s}'|\bar{s}, a) = \frac{N_{\mathcal{D}}(\bar{s}, a, \bar{s}')}{N_{\mathcal{D}}(\bar{s}, a)} \tag{9}$$

Where $N_{\mathcal{D}}(\bar{s}, a)$ denotes the number of samples in $\mathcal{D}$ where action a is taken in $\bar{s}$, and $N_{\mathcal{D}}(\bar{s}, a, \bar{s}')$ denotes the number of samples transitioning to $\bar{s}'$. The value of abstract MDP, $\rho\left(\bar{\pi}, \hat{\bar{M}}\right)$, naturally mirrors that of the original MDP. The optimal abstract policy is defined as $\bar{\pi}^* \in \arg\max_{\pi \in \Pi} \rho\left(\bar{\pi}, \hat{\bar{M}}\right)$, and can be directly learned by vanilla batch RL algorithms[18, 19, 24].

While approximate stochastic bisimulation improves sample efficiency, it inevitably introduces a performance gap between the policy learned from the abstracted dataset and that learned from the original dataset, thereby presenting challenges in providing confidence in the performance of the abstracted policy. For simplicity, we refer to approximate stochastic bisimulation as state abstraction in the remainder of this paper.

3 SPIBB with State Abstraction

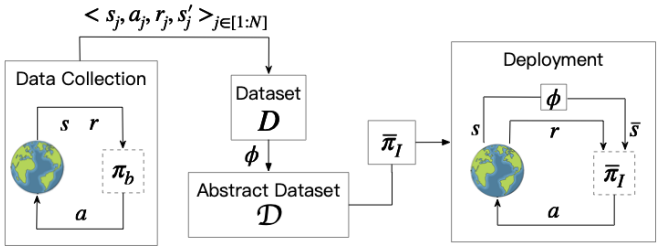

Fig.2. Illustration of the SPI with State Abstraction

Fig. 2 illustrates the framework of the SPI problem with state abstraction. It assumes the existence of an approximate stochastic bisimulation function $\phi: S \to \bar{S}$, which maps the original dataset $D = \{(s_j, a_j, r_j, s'_j)|j \in [1, N]\}$ into an abstract dataset $\mathcal{D} = \{(\bar{s}_j, a_j, r_j, \bar{s}'_j)|j \in [1, N]\}$, where $\bar{s}_j = \phi(s_j)$ and $\bar{s}'_j = \phi(s'_j)$. The objective is to design constraints that allow learning an abstract policy $\bar{\pi}_I$ from $\mathcal{D}$ such that it outperforms a behavior policy π_b in the true environment M^*. Formally, the goal is to learn a policy $\bar{\pi}_I$ that, with high probability at least $1 - \delta$, satisfies:

$$\rho(\bar{\pi}_I, M^*) \geq \rho(\pi_b, M^*) - \zeta \tag{10}$$

Where $\rho(\cdot, M^*)$ denotes the expected return (also called performance) in the true environment M^*, and ζ is admissible performance loss that quantifies the confidence level in the performance of $\bar{\pi}_I$.

3.1 SPIBB based on State Abstraction

In this section, we introduce how to learn an abstract policy $\bar{\pi}_I$ using the SPIBB algorithm based on state abstraction.

To begin with, we define the set of abstract state-action pairs with insufficient data, as directly learning a policy from such data may lead to unstable performance. To mitigate this issue, the learned policy is constrained to match the behavior policy on these pairs. Specifically, the set $\mathcal{B}$ of abstract state-action pairs is defined as:

$$\mathcal{B} = \{(\bar{s}, a) \in \bar{S} \times A | \forall \bar{s} \in \bar{S}, a \in A \; s.t. \; N_{\mathcal{D}}(\bar{s}, a) \leq N_{\wedge}\} \tag{11}$$

where $N_{\mathcal{D}}(\bar{s}, a)$ denotes the number of samples of the abstract state-action pair $(\bar{s}, a) \in \bar{S} \times A$ in the dataset $\mathcal{D}$, and $N_{\wedge}$ is a threshold parameter.

Next, we define the policy search space based on the set $\mathcal{B}$. The searchable policy space $\bar{\Pi}_b$ consists of abstract policies $\bar{\pi}_{spibb}: \bar{S} \to A$, where each policy $\bar{\pi}_{spibb}$ adheres to the abstract baseline policy $\bar{\pi}_b$ on state-action pairs with insufficient data (i.e., those in $\mathcal{B}$). Formally, the search space is defined as:

$$\bar{\Pi}_b = \left\{ \bar{\pi}_{spibb}(a|\bar{s}) = \bar{\pi}_b(a|\bar{s}) \middle| \forall \bar{s} \in \bar{S}, a \in A \; s.t. \; (\bar{s}, a) \in \mathcal{B} \right\} \tag{12}$$

Here, $\bar{\pi}_b$ denotes the abstract policy induced by the behavior policy π_b through the state abstraction mapping ϕ. A formal definition is provided in Appendix A.

Finally, the optimal policy is obtained by searching within the constrained policy space $\overline{\Pi}_b$, and is denoted by $\bar{\pi}^{\odot}_{spibb}$. Following the model-based RL paradigm, we first construct a maximum likelihood estimate of the abstract MDP $\widehat{\bar{M}} = \left(\bar{S}, A, \widehat{\bar{T}}, R, \bar{s}_0, \gamma\right)$ based on the abstract dataset $\mathcal{D} = \left\{\left(\bar{s}_j, a_j, r_j, \bar{s}'_j\right) | j \in [1, N]\right\}$. We then apply a standard policy iteration procedure to identify the optimal policy $\bar{\pi}^{\odot}_{spibb}$ within $\overline{\Pi}_b$.

Algorithm 1 Greedy Projection of $Q^{(i)}$ on $\overline{\Pi}_b$

Input: Baseline Policy $\bar{\pi}_b$
Input: Last iteration value function $Q^{(i)}$
Input: Set of bootstrapped abstracted state-action pairs $\mathcal{B}$
Input: Current abstracted state s and action set A
1: Initialize $\bar{\pi}^{(i)}_{spibb} = 0$
2: **for** $(\bar{s}, a) \in \mathcal{B}$ **do** $\bar{\pi}^{(i)}_{spibb} = \bar{\pi}_b(a|\bar{s})$
3: $\bar{\pi}^{(i)}_{spibb}\left(\bar{s}, \text{argmax}_{a|(\bar{s},a)\notin\mathcal{B}} Q^{(i)}(\bar{s}, a)\right) = \sum_{a|(\bar{s},a)\notin\mathcal{B}} \bar{\pi}_b(a|\bar{s})$
4: **Return** $\bar{\pi}^{(i)}_{spibb}$

Algorithm 1 shows the policy iteration process under the constraint imposed by $\overline{\Pi}_b$: given the Q-values from the previous iteration, the policy and Q-values for the abstract state-action pairs in $\mathcal{B}$ are kept fixed (line 2), while updates are performed only on the remaining pairs (line 3).

3.2 Theoretical Analysis

In this section, we show that the policy $\bar{\pi}^{\odot}_{spibb}$, learned via SPIBB with state abstraction, is guaranteed with high probability to perform no worse than the baseline policy π_b, up to an acceptable performance loss ζ. The primary objective of this section is to derive a theoretical bound for ζ, which quantifies the confidence in whether $\bar{\pi}_I$ outperforms π_b, and serves as a decision criterion for replacing π_b with $\bar{\pi}_I$.

Theorem 1. Let $\overline{\Pi}_b$ denote the set of abstract policies that are constrained to follow the abstract baseline policy $\bar{\pi}_b$ for all $(\bar{s}, a) \in \mathcal{B}$. Then, the optimal policy $\bar{\pi}^{\odot}_{spibb} \in \overline{\Pi}_b$ is, with probability at least $1 - \delta$, an approximate improvement over the baseline policy π_b, with an acceptable performance loss bounded by:

$$\zeta = \left(\eta|\bar{S}| + \sqrt{\frac{8}{N_\wedge} \ln \frac{|\bar{S}||A|2^{|\bar{S}|}}{\delta}}\right) \frac{2\gamma R_{max}}{(1-\gamma)^2} - \rho\left(\bar{\pi}^{\odot}_{spibb}, \widehat{\bar{M}}\right) + \rho\left(\bar{\pi}_b, \widehat{\bar{M}}\right) \qquad (13)$$

Here, $\rho\left(\bar{\pi}^{\odot}_{spibb}, \widehat{\bar{M}}\right)$ and $\rho\left(\bar{\pi}_b, \widehat{\bar{M}}\right)$ denote the performances of $\bar{\pi}^{\odot}_{spibb}$ and $\bar{\pi}_b$, respectively, in the maximum likelihood estimate abstract MDP $\widehat{\bar{M}}$.

The proof is provided in Appendix B. Given a user-defined threshold $N_\wedge$, Theorem 1 provides a bound on the acceptable performance loss ζ, ensuring that policies learned under state abstraction still enjoy performance guarantees. A key advantage of state

abstraction is that it enables sample sharing across similar states. For instance, if two original states s_0 and s_1 are both mapped to the same abstract state $\bar{s}_0$, then the sample count in the abstract dataset satisfies $N_{\mathcal{D}}(\bar{s}_0, a) = N_D(s_0, a) + N_D(s_1, a)$. This aggregation allows a safe policy update at $\bar{s}_0$ if the total count exceeds $N_\wedge$, whereas the original SPIBB framework would require each state-action pair to meet the threshold individually. However, it is generally difficult to quantify the confidence that a learned policy outperforms the baseline policy. A key contribution of this work is to derive such a confidence ζ, enabling users to assess whether the learned policy can be safely deployed in the true environment.

4 Empirical Analysis

In the previous section, we provided a confidence bound on the performance of policies learned using the SPIBB algorithm under state abstraction. In this section, we demonstrate empirically that our approach yields higher sample efficiency, achieving greater policy improvement with less data.

4.1 Experimental Setup

We design a Markov Decision Process (MDP) with 9 fine-grained states and 2 actions, abstracted into 5 abstract states. Each abstract state groups nearby locations (e.g., rooms within the same functional zone), and transitions between states within the same abstract group are constrained to have probabilities below 0.2. This setup could model environments where a robot rarely moves between positions within the same region, such as in structured indoor patrol or building inspection tasks with sparse intra-region movement.

Our evaluation covers three methods: the standard SPIBB algorithm (π_b-SPIBB), a variant incorporating state abstraction (Abstract π_b-SPIBB), and a basic reinforcement learning baseline (BasicRL) for comparison. Our implementation is based on the original SPIBB algorithm [7]. The source code and the corresponding MDP models can be found at: https://github.com/Fishee998/SPIBB_abstraction.

4.2 Performance Evaluation

The algorithms are then evaluated using the mean performance of the policies they produced.

RQ1. Can the SPIBB method incorporating state abstraction achieve greater mean performance with fewer samples?

$N_\wedge = 5$ $N_\wedge = 7$

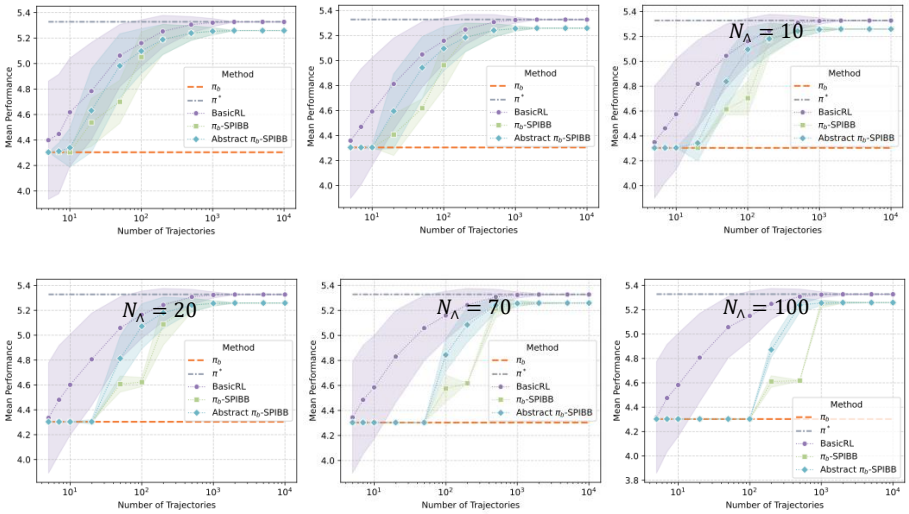

Fig.3. Mean Performance of Various Algorithms

Fig. 3 presents the mean performance comparison of various algorithms across different sample sizes and threshold values $N_\wedge$. The orange line indicates the performance of the behavior policy π_b, while π^* represents the optimal policy, which is learned by solving the true MDP model using dynamic programming. We observe that the abstracted SPIBB variant (Abstract π_b-SPIBB) consistently achieves higher or comparable mean performance to the original π_b-SPIBB method, particularly in low-data regimes (e.g., fewer than 100 trajectories). This demonstrates that incorporating state abstraction allows the algorithm to generalize from fewer samples, leading to earlier and more robust policy improvement.

As $N_\wedge$ increases (from 5 to 100), the confidence threshold for safe improvement becomes more conservative. In this case, both SPIBB variants become more cautious, but the abstracted version continues to outperform the flat SPIBB baseline in most settings, particularly when the number of trajectories is limited. Notably, Abstract π_b-SPIBB approaches the performance of the optimal policy π^* more quickly than the other baselines. These results support the hypothesis that combining SPIBB with state abstraction improves sample efficiency, enabling better policy learning under smaller data.

5 Related Works

Batch reinforcement learning (Batch RL)[4], also known as offline reinforcement learning [5], focuses on how to learn a policy from pre-collected fixed dataset when the agent cannot directly interact with the environment. This paper addresses the safety policy improvement (SPI) problem in batch RL [6-9], which involves learning a policy from a fixed dataset and guaranteeing that the learned policy outperforms the behavior policy used to generate those samples. In reinforcement learning, the concept of "safety" can have multiple meanings [25], including parameter uncertainty [26], model

uncertainty [27], external interruptibility [28, 29], and safety concerns in exploration in risky environments [30, 31]. The SPI problem primarily concerns safety related to parameter uncertainty.

Early approaches to the SPI problem mainly used the model-free RL paradigm, where policies are learned from a dataset without constructing an environment model [32, 33]. These methods work well only when nondeterministic parameters, like transition probabilities, follow a uniform distribution. Otherwise, they return the behavior policy and cannot generate the target policy. The paper [6] adopts a model-based approach, minimizing robust baseline regret, which transforms the SPI problem into a state-action pairs version, allowing it to handle nondeterministic parameters that don't follow a uniform distribution. It proves that the SPI problem is NP-hard and introduces constraints to approximate target policy learning, though scalability remains limited. Paper [7] proposes the baseline-guided SPI method (SPIBB), which builds on [6] by adding constraints for different state-action pairs, ensuring performance even when some pairs don't satisfy the constraints. Other works either improve effectiveness [21, 22] or extend SPI to more complex settings, such as partially observable MDPs [9]. A common feature of these works is their reliance on the i.i.d. assumption of fixed dataset.

6 Conclusion and Future

In this paper, we studied the problem of safe policy improvement under state abstraction in the context of batch reinforcement learning. We proposed a method that integrates state abstraction into the SPIBB framework and provided theoretical confidence bounds on the performance of the learned policy under this setting. Empirical results demonstrate that our approach achieves better policy improvement with significantly fewer samples compared to the original SPIBB method, highlighting the effectiveness of abstraction in improving sample efficiency while maintaining safety guarantees.

There are two promising directions for future research. The first is to adaptively learn the state abstraction function from different datasets, and further derive corresponding training constraints under such learned state abstraction function. The second is to extend our framework to more complex models, such as partially observable Markov decision processes (POMDPs), to improve sample efficiency.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
2. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G.: Human-level control through deep reinforcement learning. nature 518, 529-533 (2015)
3. Retzlaff, C.O., Das, S., Wayllace, C., Mousavi, P., Afshari, M., Yang, T., Saranti, A., Angerschmid, A., Taylor, M.E., Holzinger, A.: Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. Journal of Artificial Intelligence Research 79, 359-415 (2024)

4. Lange, S., Gabel, T., Riedmiller, M.: Batch reinforcement learning. Reinforcement learning: State-of-the-art, pp. 45-73. Springer (2012)
5. Jia, Z., Rakhlin, A., Sekhari, A., Wei, C.-Y.: Offline Reinforcement Learning: Role of State Aggregation and Trajectory Data. arXiv preprint arXiv:2403.17091 (2024)
6. Ghavamzadeh, M., Petrik, M., Chow, Y.: Safe policy improvement by minimizing robust baseline regret. Advances in Neural Information Processing Systems 29, (2016)
7. Laroche, R., Trichelair, P., Des Combes, R.T.: Safe policy improvement with baseline bootstrapping. In: International conference on machine learning, pp. 3652-3661. PMLR, (Year)
8. Scholl, P., Dietrich, F., Otte, C., Udluft, S.: Safe policy improvement approaches and their limitations. In: International Conference on Agents and Artificial Intelligence, pp. 74-98. Springer, (Year)
9. Simão, T.D., Suilen, M., Jansen, N.: Safe Policy Improvement for POMDPs via Finite-State Controllers. arXiv preprint arXiv:2301.04939 (2023)
10. Chandak, Y., Jordan, S., Theocharous, G., White, M., Thomas, P.S.: Towards safe policy improvement for non-stationary MDPs. Advances in Neural Information Processing Systems 33, 9156-9168 (2020)
11. Simão, T.D., Laroche, R., Tachet des Combes, R.: Safe Policy Improvement with an Estimated Baseline Policy. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1269-1277. (Year)
12. Nadjahi, K., Laroche, R., Tachet des Combes, R.: Safe policy improvement with soft baseline bootstrapping. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part III, pp. 53-68. Springer, (Year)
13. Starre, R.A., Loog, M., Congeduti, E., Oliehoek, F.A.: An Analysis of Model-Based Reinforcement Learning From Abstracted Observations. Transactions on Machine Learning Research (2023)
14. Paduraru, C., Kaplow, R., Precup, D., Pineau, J.: Model-based reinforcement learning with state aggregation. In: 8th European Workshop on Reinforcement Learning. (Year)
15. Starre, R.A., Loog, M., Oliehoek, F.A.: Model-Based Reinforcement Learning with State Abstraction: A Survey. In: BNAIC/BeNeLearn 2022. (Year)
16. Bellman, R.: A Markovian decision process. Journal of mathematics and mechanics 679-684 (1957)
17. Moerland, T.M., Broekens, J., Plaat, A., Jonker, C.M.: Model-based reinforcement learning: A survey. Foundations and Trends® in Machine Learning 16, 1-118 (2023)
18. Chatterjee, K., Henzinger, T.A.: Value iteration. 25 Years of Model Checking: History, Achievements, Perspectives, pp. 107-138. Springer (2008)
19. Watkins, C.J., Dayan, P.: Q-learning. Machine learning 8, 279-292 (1992)
20. Auer, P., Jaksch, T., Ortner, R.: Near-optimal regret bounds for reinforcement learning. Advances in neural information processing systems 21, (2008)
21. Li, L., Walsh, T.J., Littman, M.L.: Towards a unified theory of state abstraction for MDPs. In: AI&M. (Year)
22. Ortner, R., Maillard, O.-A., Ryabko, D.: Selecting near-optimal approximate state representations in reinforcement learning. In: International Conference on Algorithmic Learning Theory, pp. 140-154. Springer, (Year)

23. Abel, D., Arumugam, D., Lehnert, L., Littman, M.: State abstractions for lifelong reinforcement learning. In: International Conference on Machine Learning, pp. 10-19. PMLR, (Year)
24. Ernst, D., Geurts, P., Wehenkel, L.: Tree-based batch mode reinforcement learning. Journal of Machine Learning Research 6, (2005)
25. Garcıa, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research 16, 1437-1480 (2015)
26. Thomas, P., Theocharous, G., Ghavamzadeh, M.: High confidence policy improvement. In: International Conference on Machine Learning, pp. 2380-2388. PMLR, (Year)
27. Altman, E.: Constrained Markov decision processes. Routledge (2021)
28. El Mhamdi, E.M., Guerraoui, R., Hendrikx, H., Maurer, A.: Dynamic safe interruptibility for decentralized multi-agent reinforcement learning. Advances in Neural Information Processing Systems 30, (2017)
29. Orseau, L., Armstrong, M.: Safely interruptible agents. In: Conference on Uncertainty in Artificial Intelligence. Association for Uncertainty in Artificial Intelligence, (Year)
30. Schulman, J.: Trust Region Policy Optimization. arXiv preprint arXiv:1502.05477 (2015)
31. Fatemi, M., Sharma, S., Van Seijen, H., Kahou, S.E.: Dead-ends and secure exploration in reinforcement learning. In: International Conference on Machine Learning, pp. 1873-1881. PMLR, (Year)
32. Thomas, P., Theocharous, G., Ghavamzadeh, M.: High-confidence off-policy evaluation. In: Proceedings of the AAAI Conference on Artificial Intelligence. (Year)
33. Kakade, S., Langford, J.: Approximately optimal approximate reinforcement learning. In: Proceedings of the Nineteenth International Conference on Machine Learning, pp. 267-274. (Year)

Appendix

A. Definition of the Abstract Behavior policy

We define an abstract behavior policy $\bar{\pi}_b$ that preserves the action choices of the original behavior policy π_b under the ε-bisimulation function $\phi: S \to \bar{S}$. Specifically, for any abstract state $\bar{s} \in \bar{S}$, $\bar{\pi}_b(\bar{s})$ selects an action consistent with π_b over the set of original states mapped to $\bar{s}$, i.e.,

$$\forall \bar{s} \in \bar{S}, \bar{\pi}_b(\bar{s}) = \begin{cases} a_i, if\ s_i \in \phi^{-1}(\bar{s})\ and\ \pi_b(s_i) = a_i \\ a_j, if\ s_j \in \phi^{-1}(\bar{s})\ and\ \pi_b(s_j) = a_j \\ \cdots \end{cases}$$

By construction, $\bar{\pi}_b$ can be viewed as the abstraction of π_b in the abstract MDP. Consequently, $\bar{\pi}_b$ and π_b are performance-equivalent in their corresponding models.

B. Proof of Theorem 1

To begin with, we introduce two lemmas that are required for Theorem 1.

Lemma 1. For any abstract policy $\bar{\pi} \in \bar{\Pi}_b$, the performance difference between its execution in the abstract MDP $\bar{M}_\omega^*$ and in the true MDP M^* is bounded as follows:

$$|\rho(\bar{\pi}, \bar{M}_\omega^*) - \rho(\bar{\pi}, M^*)| \leq \frac{\gamma\eta|\bar{S}|R_{max}}{(1-\gamma)^2} \tag{14}$$

Proof. Follows identically to the proof of Lemma 6 in [10].

Lemma 2. For any abstract policy $\bar{\pi} \in \bar{\Pi}_b$, the performance difference between its evaluation in the estimated abstract MDP $\bar{M}_\omega^*$ and in the abstract MDP M^* satisfies the following bound：

$$\left|\rho(\bar{\pi}, \bar{M}_\omega^*) - \rho\left(\bar{\pi}, \widehat{\bar{M}}\right)\right| \leq \frac{\gamma R_{max}}{(1-\gamma)^2}\sqrt{\frac{8}{N_\wedge}\ln\frac{|\bar{S}||A|2^{|\bar{S}|}}{\delta}} \tag{15}$$

Proof. Follows identically to the proof of Lemma 8 and Theorem 2 in [10].

Subsequently, using equations (14) and (15), we can derive that for any abstract policy $\bar{\pi} \in \bar{\Pi}_b$, the discrepancy in performance between the true MDP M^* and the estimated abstract MDP $\widehat{\bar{M}}$ is upper bounded by:

$$\left|\rho(\bar{\pi}, M^*) - \rho\left(\bar{\pi}, \widehat{\bar{M}}\right)\right| \leq \left(\eta|\bar{S}| + \sqrt{\frac{8}{N_\wedge}\ln\frac{|\bar{S}||A|2^{|\bar{S}|}}{\delta}}\right)\frac{\gamma R_{max}}{(1-\gamma)^2} \tag{16}$$

Substituting $\bar{\pi}_{spibb}^{\odot}$ and $\bar{\pi}_b$ into equation (16) yields the following:

$$\left|\rho\left(\bar{\pi}_{spibb}^{\odot}, M^*\right) - \rho\left(\bar{\pi}_{spibb}^{\odot}, \widehat{\bar{M}}\right)\right| \leq \left(\eta|\bar{S}| + \sqrt{\frac{8}{N_\wedge}\ln\frac{|\bar{S}||A|2^{|\bar{S}|}}{\delta}}\right)\frac{\gamma R_{max}}{(1-\gamma)^2} \tag{17}$$

$$\left|\rho(\bar{\pi}_b, M^*) - \rho\left(\bar{\pi}_b, \widehat{\bar{M}}\right)\right| \leq \left(\eta|\bar{S}| + \sqrt{\frac{8}{N_\wedge}\ln\frac{|\bar{S}||A|2^{|\bar{S}|}}{\delta}}\right)\frac{\gamma R_{max}}{(1-\gamma)^2} \tag{18}$$

By adding both sides of equations (17) and (18), we obtain:

$$\rho\left(\bar{\pi}_{spibb}^{\odot}, M^*\right) - \rho(\bar{\pi}_b, M^*) \geq \rho\left(\bar{\pi}_{spibb}^{\odot}, \widehat{\bar{M}}\right) - \rho\left(\bar{\pi}_b, \widehat{\bar{M}}\right) - \left(\eta|\bar{S}| + \sqrt{\frac{8}{N_\wedge}\ln\frac{|\bar{S}||A|2^{|\bar{S}|}}{\delta}}\right)\frac{2\gamma R_{max}}{(1-\gamma)^2} \tag{19}$$

Then, the acceptable performance loss ζ can be defined as:

$$\zeta = \left(\eta|\bar{S}| + \sqrt{\frac{8}{N_\wedge}\ln\frac{|\bar{S}||A|2^{|\bar{S}|}}{\delta}}\right)\frac{2\gamma R_{max}}{(1-\gamma)^2} - \rho\left(\bar{\pi}_{spibb}^{\odot}, \widehat{\bar{M}}\right) + \rho\left(\bar{\pi}_b, \widehat{\bar{M}}\right) \tag{20}$$

We have

$$\rho\left(\bar{\pi}_{spibb}^{\odot}, M^*\right) \geq \rho(\bar{\pi}_b, M^*) - \zeta$$

This concludes the proof.