State Proficiency-Based Adaptive Fine-Tuning for Offline-to-Online Reinforcement Learning

Songlin Li[1,2], **Wei Xiao**[1,2], **Hao Wu**[1,2], **Xiaodan Zhang**[1,2], **Daolong An**[1,2], **Shuai Lü**[1,2,3*]

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, China
[2]College of Computer Science and Technology, Jilin University, China
[3]College of Software, Jilin University, China
lus@jlu.edu.cn, {lisl23, weixiao23, haowu24, xdzhang24, andl22}@mails.jlu.edu.cn

Abstract

In offline-to-online (O2O) reinforcement learning, achieving efficient performance improvement while maintaining training stability remains a critical challenge for effective fine-tuning. Existing O2O methods usually focus on the balance between policy improvement and policy constraint during online fine-tuning. However, they often overlook sample differences, leading to suboptimal performance. To address this challenge, we identify that the effectiveness of policy learning exhibits significant variation across states. Therefore, we propose the notion of state proficiency to capture the degree of effective learning in a given state. We propose State Proficiency-Based Adaptive Fine-Tuning (SPA), a straightforward yet effective method that establishes proficiency-based sample priorities in policy optimization to facilitate effective fine-tuning. Specifically, SPA focuses on low proficiency samples during policy improvement to enhance sample efficiency, while emphasizing high proficiency samples during policy constraint to ensure stable training. Extensive empirical results demonstrate that SPA achieves significant improvements over existing methods, attaining state-of-the-art performance on the D4RL benchmark.

Introduction

Offline reinforcement learning (RL) focuses on learning optimal policy from the fixed dataset, eliminating the need for environmental interaction (Levine et al. 2022). It is particularly valuable in high-risk scenarios such as healthcare (Tang et al. 2022) and robotics (Diehl et al. 2023). However, the performance of offline policy is often constrained by the quality of dataset (Luo et al. 2024). The offline-to-online (O2O) RL paradigm addresses this limitation by enhancing policy performance through online fine-tuning (Nair et al. 2020; Lee et al. 2021). Although both online fine-tuning and traditional online RL learn through environmental interaction, two fundamental discrepancies exist among these training paradigm: initial policy performance and training objective. Fine-tuning focuses on pre-trained agent with reliable performance, aiming to efficiently improve its performance with limited interactions. While online RL starts from randomly initialized agent, which typically exhibits poor initial performance. It requires extensive environmental interactions for exploration, resulting in low sample efficiency. Furthermore, fine-tuning should maintain characteristics applicable to high-risk environments similar to offline RL, the training stability is considerable throughout the fine-tuning process. Therefore, achieving both efficient performance improvement and stable training is essential for effective online fine-tuning.

Conventional O2O RL algorithms face challenges of training instability and slow performance improvement (Nakamoto et al. 2023; Kong et al. 2024). For instance, directly applying online algorithms (Haarnoja et al. 2018; Fujimoto, Hoof, and Meger 2018) to O2O suffers from severe performance fluctuations during fine-tuning, as the aggressive policy optimization amplifies the distribution shift (Lee et al. 2021; Zhang et al. 2024; Zhao et al. 2024). Adopting offline algorithms (Fujimoto and Gu 2021; Kostrikov, Nair, and Levine 2022) can mitigate distribution shift with policy constraint, but excessive conservatism hinders efficient performance improvement (Nair et al. 2020; Nakamoto et al. 2023). Prior O2O algorithms (Nair et al. 2020; Zhao et al. 2022; Wang et al. 2023) aim to achieve effective fine-tuning by establishing a rational balance between policy improvement and policy constraint during policy optimization. However, recent algorithms (Wu et al. 2022; Lyu et al. 2022) often overlook sample differences, resulting in a uniform balance applied across all samples. To address this issue, we propose the notion of state proficiency, a sample-level characteristics which integrates composite information about state and policy action, capturing how effectively a policy learns in a given state. We identify significant variation in state proficiency across states. The core insight lies in the integration of state proficiency with sample-level balance, thereby establishing effective fine-tuning.

In this paper, we propose State Proficiency-Based Adaptive Fine-Tuning (SPA), a novel method that establishes proficiency-based sample priorities in policy optimization to facilitate effective fine-tuning. Specifically, SPA focuses on low proficiency samples during policy improvement to facilitate efficient performance improvement, while emphasizing high proficiency samples during policy constraint to ensure stable training. Our algorithm consists of two main processes: state proficiency assessment and proficiency-based adaptive policy constraint. To efficiently and accu-

*Corresponding author.

rately determine the state proficiency, we propose an action-comparison-based classification, along with dual thresholds and value-function-based classification correction. Following assessment results, we propose an effective adaptive policy constraint based on proficiency levels to establish adaptive fine-tuning. To our knowledge, SPA is the first O2O RL approach that leverages the state proficiency for adaptive fine-tuning.

The main contributions of this paper are as follows:

- We propose the notion of state proficiency, designing proficiency-based sample priorities during policy optimization for effective fine-tuning.
- We propose a practical O2O RL algorithm, SPA, which incorporates a rational proficiency assessment, along with adaptive policy constraint based on proficiency.
- Empirical results demonstrate that our proposed algorithm achieves state-of-the-art (SOTA) performance across multiple environments and datasets.

Preliminaries

Reinforcement Learning

RL is typically expressed as a Markov decision process (MDP) (Sutton and Barto 2018), denoted as $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the environmental state transition probability, R is the reward function and γ is a discount factor. The objective of RL is to learn a policy π that maximizes the discounted cumulative reward as $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$.

The corresponding state-action value function Q^π and state value function V^π are the expected return: $Q^\pi(s,a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a]$, and $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s] = \mathbb{E}_{a\sim\pi}[Q^\pi(s,a)]$.

Offline-to-Online RL

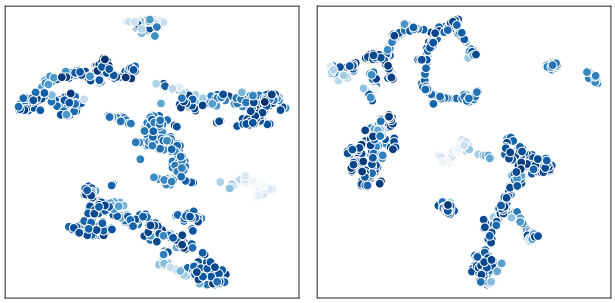
The training objective of policy optimization in O2O RL typically consists of two primary components: policy improvement and policy constraint. Policy improvement maximizes current value function to guide policy learning. Policy constraint restricts the policy to remain close to the distribution of the dataset or the current replay buffer. Achieving an effective balance between policy improvement and policy constraint is a central focus of O2O research (Wang et al. 2023). For clarity and analysis, we employ TD3+BC (Fujimoto and Gu 2021) as the backbone algorithm. The loss function of policy training in TD3+BC is as follows:

$$L_\pi(\phi) = \mathbb{E}_{(s,a)\sim D}[-\lambda Q_\theta(s, \pi_\phi(s)) + (\pi_\phi(s) - a)^2], \quad (1)$$

where D represents the collected dataset, π_ϕ and Q_θ denote the learning policy and the corresponding state-action value function, parameterized by ϕ and θ, respectively, $\lambda = \alpha N / \sum_{s_i,a_i} Q(s_i, a_i)$, α is a hyperparameter and N is the batch size.

Differences in State Proficiency Levels

Stemming from dataset limitations and inherent constraints in policy generalization capabilities, the effectiveness of offline pre-training varies significantly across states. Therefore, we propose the notion of **state proficiency**, which serves as a characteristic of a state to capture the effectiveness of policy learning in that state. For a given state, the more effectively the policy learns in that state, the higher its state proficiency is. For instance, in states out of dataset distribution or in-distribution states with suboptimal dataset action, the policy exhibits inadequate learning, resulting in unsatisfactory performance. The state proficiency level of such sample is relatively low; Conversely, states well-covered in the dataset with high quality actions yield effective policy learning, which indicates a high state proficiency level.



(a) Offline Samples (b) Online Samples

Figure 1: t-SNE visualization of state proficiency on hopper-medium task. For both online and offline samples, darker colors signify higher action quality, indicating higher state proficiency.

We validate the state proficiency variation through empirical experiments. The core insight lies in the intrinsic correlation between the action quality of the learning policy and the state proficiency level. Higher proficiency levels correspond to superior action. Therefore, we employ the true Q-values determined through Monte Carlo (MC) rollouts (Sutton and Barto 2018) to represent the action quality for measuring state proficiency. We select the hopper-medium dataset in D4RL (Fu et al. 2020) and visualize the state proficiency levels of the offline pre-trained policy of TD3+BC. Offline states are obtained by random sampling from the dataset, while online states are sampled through environmental interactions using the pre-trained policy. We utilize t-SNE (Maaten and Hinton 2008) to visualize proficiency levels in Figure 1, with color coding indicating the true Q-values of policy actions at each state.

Results demonstrate significant variation in proficiency levels across different states, both in offline states with distribution shift and in online states closely related to the policy. This inherent disparity motivates us to investigate adaptive fine-tuning based on state proficiency.

Method

Motivated by the variation of state proficiency across different samples, we propose State Proficiency-Based Adaptive Fine-Tuning (SPA) for effective fine-tuning. The full algorithm is summarized in Algorithm 1. SPA consists of two components: **state proficiency assessment** and **proficiency-based adaptive policy constraint**. Specifically, with the the state proficiency assessment in lines 8–9, the samples within a mini-batch are categorized into low and high proficiency

Algorithm 1: SPA

Require: Pre-trained actor π_{offline}, Q-value function Q_{offline}, value function V_{offline}, offline dataset D_{offline} and hyperparameter η

1: Initialize empty online buffer D_{online}, π^0, Q^0 and V^0
2: Set the historical optimal policy $\pi_{\text{opt}} \leftarrow \pi^0$
3: **for** $t = 1, 2, \cdots$ **do**
4: Collect samples with π and store samples in D_{online}
5: Sample a mini-batch B from $D_{\text{online}} \cup D_{\text{offline}}$
6: Update Q with B according to the base algorithm
7: Update V with B by Eq. (4)
8: Divide B into S_{HP} and S_{LP} by Eqs. (2) and (3)
9: Update S_{HP} and S_{LP} by Eq. (5)
10: Get the adaptive weight ω_s for each sample in B with S_{HP} and S_{LP} by Eq. (7)
11: Update π with B by Eq. (6)
12: **end for**

categories during each training iteration. Based on the classification results, SPA optimizes the learning policy through adaptive policy constraint in lines 10–11. In this section, we first present the core insight of our approach. Then we detail the state proficiency assessment within SPA. Finally we describe the proficiency-based adaptive policy constraint.

Leveraging State Proficiency for Effective Fine-Tuning

The objective of policy optimization consists of policy improvement and policy constraint. Our core insight lies in establishing proficiency-based sample priorities in policy optimization to facilitate effective fine-tuning. Specifically:

- SPA achieves efficient performance improvement by increasing the weights of low proficiency samples in policy improvement objective. It prioritizes maximizing the Q-values of low proficiency samples and enhancing the policy performance on these samples, achieving superior overall performance.
- SPA ensures stable training by enhancing the weights of high proficiency samples in policy constraint objective. It prioritizes the constraint on high proficiency samples and focuses on maintaining performance stability for these samples, resulting in more stable training.

State Proficiency-based Sample Classification through Action Comparison

We propose a simple yet effective proficiency-based sample classification through action comparison. We employ a reference policy and a Q-value function to construct a threshold function for state proficiency classification. Specifically, we first establish a reference policy that exhibits reliable performance. It emphasizes that reference policy π_{ref} should serve as a meaningful baseline for proficiency assessment. Concretely, π_{ref} should at least outperform a random policy. Second, we utilize a Q-value function Q that accurately estimates the action value. The threshold function $\sigma(s)$ for state proficiency classification can be formally defined as $\sigma(s) = Q(s, \pi_{\text{ref}}(s))$. Comparing the Q-value of learning policy π to the threshold allows us to classify the samples within a mini-batch into high and low proficiency categories. For each state s in a mini-batch of states S, if $Q(s, \pi(s))$ exceeds $\sigma(s)$, state s is classified into high state proficiency sample set S_{HP}; Otherwise, it is categorized into low state proficiency sample set S_{LP}. It can be formalized as:

$$S_{\text{HP}} = \{s \in S \mid Q(s, \pi(s)) \geq \sigma(s)\} \text{ and } S_{\text{LP}} = S - S_{\text{HP}}. \tag{2}$$

The central premise behind action comparison is that the action quality can adequately reflect the state proficiency levels, and the higher Q-value indicates superior action quality in a given state. We exploit the continuously updated Q-value function Q^π during fine-tuning for proficiency-based classification. Q^π is typically initialized with the offline Q-value function, ensuring that its value estimations are reasonably accurate. As the fine-tuning progresses, the estimate accuracy of Q^π improves. This allows us to avoid introducing the additional models and extra computational costs. More disucission Q-value accuracy requirements for SPA is provided in Appendix.

Dual Classification Thresholds

Consistent with established practices in O2O RL (Nair et al. 2020; Zheng et al. 2023), we utilize both offline and online data during fine-tuning. The discrepancy among data sources should be explicitly considered in proficiency assessment. Therefore, we propose dual thresholds for state proficiency assessment: Using the frozen offline policy π_{off} to derive the relatively low threshold for D_{offline} samples; Adopting the historical optimal policy π_{opt} to establish the relatively high threshold for online samples. The policy selection is based on data source and the threshold calculation is hyperparameter-free. The configuration of π_{opt} is aligned with OCR (Luo et al. 2024). Threshold function $\sigma(s)$ with dual thresholds is formulated as follows:

$$\sigma(s) = \begin{cases} Q^\pi(s, \pi_{\text{off}}(s)), & \text{if } s \in D_{\text{offline}}, \\ Q^\pi(s, \pi_{\text{opt}}(s)), & \text{otherwise.} \end{cases} \tag{3}$$

The dual thresholds design stems from the inherent differences between offline and online data. Offline data, collected from the behavior policy π_β, is employed for policy optimization during offline pre-training. The learning policy π typically exhibits reliable performance with these states, necessitates the conservative policy learning on offline samples for enabling fine-tuning stability. In contrast, online data generated by π establishes a closer connection with π and contains real-time feedback. It is enriched with more informative signals, warranting aggressive learning for efficient performance improvement and facilitating error correction.

To leverage these properties, we establish distinct thresholds. The relatively low threshold from π_{off} increases proportion of high proficiency samples within offline data, promoting more conservative optimization to stabilize performance. While the relatively high threshold derived from π_{opt} increases proportion of low proficiency samples within online data, adopting more aggressive training to facilitate performance improvement.

Value-based Classification Correction

However, the overestimation bias of Q^π (Fujimoto, Hoof, and Meger 2018) inevitably skews the state proficiency determination. Specifically, Q^π may overestimate the action quality of learning policy π or underestimate the threshold $\sigma(s)$, misclassifying actual low proficiency samples as high proficiency. This inflates high proficiency sample proportion, fostering overly conservative training and suboptimal convergence, ultimately degrading performance.

To mitigate this challenge, we introduce a state value function V_ψ^π, parameterized by ψ for classification correction. During fine-tuning, V_ψ^π approximates the expected value solely concerning the Q-function, and can accurately estimate the average action quality in a given state. The loss function for training can be formulated as follows:

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\left(Q_\theta^\pi(s,a) - V_\psi^\pi(s)\right)^2\right]. \quad (4)$$

In our proficiency-based sample classification correction, within the subset of initially classified high proficiency samples, we compare the action quality of π against the average action quality estimated by V^π. If the action quality still exceeds the new threshold, we retain the original classification; Otherwise, we infer that estimation bias may mislead the classification and correct these misclassified samples to low proficiency category. The classification correction can be formalized as follows:

$$\begin{aligned} S_{\text{HP}} &\leftarrow S_{\text{HP}} - \{s \in S_{\text{HP}} \mid Q^\pi(s,\pi(s)) < V^\pi(s)\}, \\ S_{\text{LP}} &\leftarrow S_{\text{LP}} \cup \{s \in S_{\text{HP}} \mid Q^\pi(s,\pi(s)) < V^\pi(s)\}. \end{aligned} \quad (5)$$

The essence of classification correction lies in V^π capturing the average action quality of the learning policy π rather than the reference policy π_{ref}. This distinction eliminates the potential underestimation of classification threshold. Additionally, the training alignment of V^π with Q^π may help mitigate the overestimations of Q^π related to π.

State Proficiency-Based Adaptive Policy Constraint

Following the state proficiency assessment, we propose a practical state proficiency-based adaptive policy constraint to establish sample priorities in policy optimization.

Considering the sample priorities, the straightforward implementation is dynamically assigning policy improvement weight ω_s^E and policy constraint weight ω_s^C to each sample s. Specifically, for any high state proficiency sample s_h and low state proficiency sample s_l, the training object of policy improvement is $\mathcal{L}^E = \omega_{s_h}^E \mathcal{L}_{s_h}^E + \omega_{s_l}^E \mathcal{L}_{s_l}^E$, with $\omega_{s_h}^E < \omega_{s_l}^E$. For policy constraint, we have $\mathcal{L}^C = \omega_{s_h}^C \mathcal{L}_{s_h}^C + \omega_{s_l}^C \mathcal{L}_{s_l}^C$, with $\omega_{s_h}^C > \omega_{s_l}^C$. Given the policy optimization objective consists solely of policy improvement and policy constraint. Therefore, in our practical implementation, we propose an adaptive weighting $\omega_s := \omega_s^C / \omega_s^E$. It means that for any $s_h \in S_{\text{HP}}$ and $s_l \in S_{\text{LP}}$, we have $\omega_{s_h} > \omega_{s_l}$. To avoid the overly complex designs for the value of ω_s, we simplify the determination of ω_s as: Assigning a fixed value η $(\eta > 0)$ for high state proficiency samples, where η is a hyperparameter; While setting $\omega_s = 0$ for low state proficiency samples. The final loss function for our policy training is as follows:

$$L_\pi^{\text{SPA}}(\phi) = \mathbb{E}_{(s,a)\sim D}\left[-\lambda Q_\theta(s,\pi_\phi(s)) + \omega_s(\pi_\phi(s)-a)^2\right]. \quad (6)$$

$$\omega_s = \eta \cdot \mathbb{I}[\, s \in S_{\text{HP}}\,]. \quad (7)$$

Our adaptive policy constraint establishes sample-level balance between policy improvement and policy constraint. For any sample with low state proficiency, the balance prioritizes policy improvement, whereas high proficiency samples receive stronger policy constraint.

During fine-tuning, low proficiency samples benefit from aggressive learning to efficiently improve performance. Upon achieving high proficiency level, these samples shift to conservative learning to stabilize their performance. Meanwhile, for samples experiencing proficiency degradation due to errors, aggressive learning can swiftly restore their performance, enabling them to regain high proficiency level. This dynamic balancing mechanism ensures continuous alignment between the optimized policy π with the reference policy π_{ref}. Employing progressively improved π_{opt} as π_{ref} for online samples, the dynamic optimization process effectively guides π to progressively approach higher-level policy, thereby maintaining training stability and improving overall performance to achieve effective fine-tuning.

Experiments

Experiment Settings

Datasets. Our experimental evaluation employs D4RL benchmark (Fu et al. 2020) covering Locomotion and Maze2D navigation tasks. The Locomotion domain provides three environments: hopper, walker2d, and halfcheetah, each containing four quality levels. The Maze2D benchmark includes six distinct configurations.

Training Protocol. All algorithms undergo 1M training steps for offline pre-training while conduct 0.3M steps for fine-tuning. Experimental results are averaged over 5 random seeds. More implementation details refer to Appendix.

Baselines. We establish baseline comparisons with SOTA algorithms across three categories: (i) **offline-to-online RL**, includes PEX (Zhang, Xu, and Yu 2023), Cal-QL (Nakamoto et al. 2023), FamO2O (Wang et al. 2023), CPR (Kong et al. 2024) and O2TD3 (Luo et al. 2024); (ii) **online RL**, TD3 (Fujimoto, Hoof, and Meger 2018); (iii) **offline RL**, TD3+BC (Fujimoto and Gu 2021). Consistent with official implementations, FamO2O and PEX employ IQL (Kostrikov, Nair, and Levine 2022) for offline training, while Cal-QL utilizes CQL (Kumar et al. 2020) for pre-training. TD3+BC serves as the offline algorithm for other baselines and our method. Code is available at https://github.com/mikiya1213/SPA.

Benchmark Comparison

Figure 2 shows the learning curves on Locomotion during fine-tuning. Results indicate that SPA outperforms baselines in most tasks, achieving both the highest final performance and stable training process. While TD3 occasionally exhibits competitive performance, its significant high-variance training dynamics lead to suboptimal performance. Notably, SPA

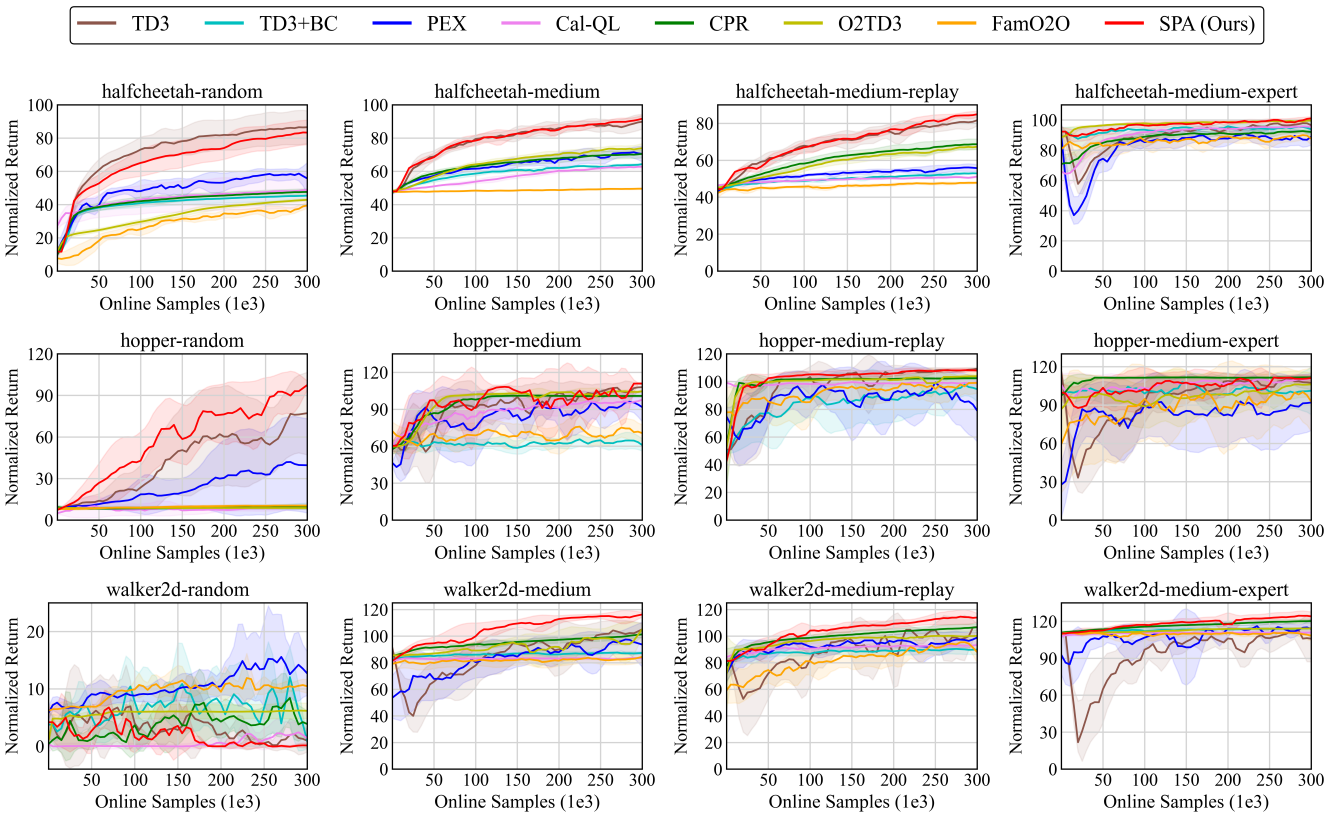


Figure 2: Performance curves on D4RL Locomotion benchmark during online fine-tuning.

Dataset	PEX	Cal-QL	CPR	O2TD3	FamO2O	SPA (Ours)
halfcheetah-random-v2	13.3→ 54.6	25.6 → 48.9	11.6 → 47.7	11.0 → 42.9	8.2 → 40.0	11.7→**83.3**
hopper-random-v2	8.2 → 39.5	7.6 → 12.0	9.3 → 9.4	8.1 → 8.7	8.1 → 10.4	8.5 → **97.7**
walker2d-random-v2	5.8 → 12.7	0.4 → 1.2	0.3 → 3.9	0.9 → 6.2	6.2 → 10.5	4.2 → 0.2
halfcheetah-medium-v2	48.2 → 70.0	47.8 → 62.9	47.6 → 70.6	48.2 → 74.5	47.9 → 49.6	48.0 → **91.7**
hopper-medium-v2	60.9 → 89.9	60.2 → 97.5	58.2 → 100.8	60.0 → 104.5	69.7 → 71.8	60.3 → **110.9**
walker2d-medium-v2	74.8 → 91.7	80.3 → 83.2	85.5 → 100.9	85.7 → 104.6	77.2 → 84.3	83.2 → **116.1**
halfcheetah-medium-replay-v2	43.5 → 55.9	46.4 → 51.3	44.4 → 68.6	44.9 → 67.4	42.2 → 47.7	44.4 → **84.7**
hopper-medium-replay-v2	76.9 → 86.1	99.1 → 99.1	57.5 → 102.4	46.2 → 103.9	84.2 → 101.8	51.0 → **107.7**
walker2d-medium-replay-v2	72.8 → 98.8	84.4 → 93.3	78.9 → 106.8	67.7 → 100.0	59.1 → 88.2	81.3 → **113.8**
halfcheetah-medium-expert-v2	90.7 → 90.1	67.6 → 95.4	73.4 → 92.5	89.9 → 99.7	80.0 → 87.3	92.3 → **100.7**
hopper-medium-expert-v2	38.9 → 97.4	105.7 → 109.8	97.7 → 111.7	81.8 → 104.6	48.1 → 86.1	99.3 → **111.8**
walker2d-medium-expert-v2	109.0 → 113.0	108.1 → 111.0	111.0 → 120.2	110.2 → 114.3	109.4 → 108.9	110.2 → **124.1**
Locomotion Total	643.0 → 899.7	733.2 → 865.6	675.4 → 935.5	654.6 → 931.3	640.3 → 786.6	694.4 → **1142.7**
Locomotion Improvement (%)	39.9	18.1	38.5	42.3	22.8	**64.6**
maze2d-umaze-v1	53.9 → 155.4	-	43.0 → 127.6	37.8 → 165.7	48.5 → 147.7	43.6 → **166.9**
maze2d-medium-v1	90.4 → 177.7	-	66.6 → 147.4	31.4 → 181.1	41.8 → 150.0	64.2 → **184.1**
maze2d-large-v1	66.6 → 187.6	-	69.3 → 161.9	123.9 → 180.6	73.4 → 150.6	90.7 → **235.0**
maze2d-umaze-dense-v1	56.5 → 123.4	-	28.6 → 87.5	35.8 → 150.0	44.5 → 85.8	35.6 → **152.6**
maze2d-medium-dense-v1	70.6 → 138.8	-	43.8 → 95.0	43.8 → 163.9	52.4 → 103.1	55.6 → **171.2**
maze2d-large-dense-v1	76.2 → 169.6	-	70.9 → 152.9	63.6 → 176.7	89.0 → 159.1	72.7 → **206**
Maze2D Total	414.2 → 952.5	-	322.2 → 772.3	336.3 → 918.0	349.6 → 896.3	362.4 → **1115.8**
Maze2D Improvement (%)	130.0	-	139.7	173.0	156.4	**207.9**

Table 1: Normalized scores before & after online fine-tuning on D4RL tasks.

exhibits a training failure on walker2d-random, which we attribute to Q-value explosion occurring during offline pre-training, thus corrupting the state proficiency assessment.

Table 1 presents the offline pre-training performance and

Dataset (-v2)	SPA	random-SPA
halfcheetah-m	**91.7**	60.3
hopper-m	**110.9**	105.3
walker2d-m	116.1	**116.3**
halfcheetah-mr	84.7	**84.8**
hopper-mr	**107.7**	106.7
walker2d-mr	**113.8**	106.7
halfcheetah-me	**100.7**	100.3
hopper-me	**111.8**	110.1
walker2d-me	**124.1**	119.6
Total	**961.6**	910.1

Table 2: Ablation study on the action-comparison-based proficiency assessment.

Dataset (-v2)	Online data	Offline data
halfcheetah-m	34.91	62.51
hopper-m	31.37	45.02
walker2d-m	35.46	52.07
halfcheetah-mr	33.79	48.07
hopper-mr	29.49	44.52
walker2d-mr	31.01	46.66
halfcheetah-me	34.51	56.61
hopper-me	30.25	33.08
walker2d-me	32.97	48.98

Table 3: The policy constraint ratios (%) of SPA.

the final fine-tuning performance, excluding Maze2D results of Cal-QL due to its complete training failure under default settings. Performance improvement rate is calculated by measuring the relative performance improvement. SPA achieves superior performance across 17/18 tasks with the highest improvement rate, establishing the SOTA performance through substantial performance margins compared to existing algorithms. The Maze2D results further validate the superiority of SPA, where it outperforms baselines in all tasks. It demonstrates the ability of SPA to stitch sub-trajectories in sparse-reward environments, expands the evidence for the adaptability of SPA across diverse task domains.

Random Proficiency Assessment

To validate the contribution of proposed state proficiency assessment, we perform ablation analysis and introduce a variant of SPA: random-SPA. In random-SPA, samples are randomly classified as high proficiency and low proficiency based on a predefined probability. To prevent various constraint ratio from interfering with the proficiency assessment methods comparison, the random probability in random-SPA matches the high proficiency ratio of SPA. This controlled setup maintains equivalent policy constraint ratio across both methods, while other training configurations remain unchanged.

As evidenced in Table 2, SPA consistently outperforms its randomized variant, underscoring the critical contribution of proficiency state assessment based on action comparisons to effective fine-tuning.

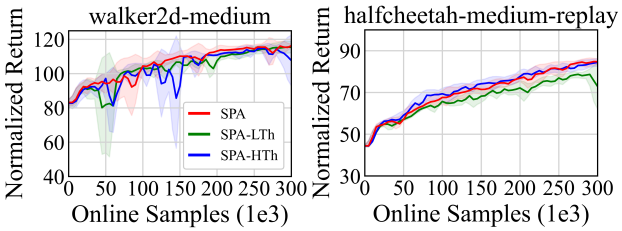


Figure 3: Ablation study on dual thresholds.

Dataset (-v2)	Data	SPA	SPA-LTh	SPA-HTh
walker2d-m	online	35.46	50.18	36.04
	offline	52.07	56.52	43.01
halfcheetah-mr	online	33.79	59.27	34.48
	offline	48.07	56.68	37.79

Table 4: The policy constraint ratios (%) on offline and online data.

Dual Thresholds vs. Single Threshold

In this section, we conduct the ablation study on the dual thresholds mechanism to evaluate its significant contribution to effective fine-tuning.

Table 3 quantifies the distinct conservative optimization effects between online and offline data through policy constraint ratios. Measured as the proportion of samples undergoing policy constraint during policy optimization per mini-batch, offline data exhibits higher constraint ratios than online counterparts. This empirical evidence confirms that the dual thresholds successfully establish a training distinction: Conservative policy training on offline data and more aggressive learning on online data.

To further investigate the essential contribution of dual thresholds to state proficiency assessment, we remove the dual thresholds and introduce two variants:

- **SPA-LTh** employs the single proficiency assessment threshold from the frozen offline policy π_{off} for both offline and online data.
- **SPA-HTh** utilizes the historical optimal policy π_{opt} to derive the single threshold for all data.

Figure 3 compares their normalized scores with SPA, and the policy constraint ratios are reported in Table 4.

Experimental results demonstrate the superiority of SPA in training stability and convergence performance over both variants. Specifically, SPA-HTh, as shown in Figure 3a, exhibits training instability and performance degradation in later training phases. This phenomenon stems from the fact that the threshold derived from the historical optimal policy π_{opt} is too high for offline data, resulting in an extremely low proportion of high proficiency samples. The resultant insufficient constraint ratio in policy optimization, as evidenced by Table 4, fundamentally destabilizes the training

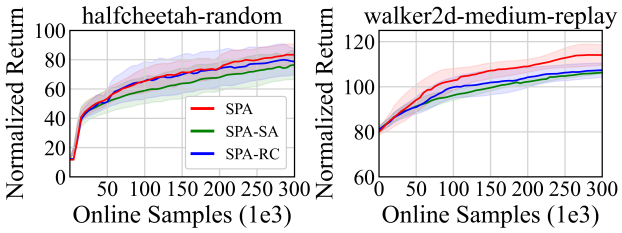


Figure 4: Ablation study on value-based classification correction.

process. Conversely, SPA-LTh exhibits suboptimal performance in Figure 3b, which we attribute to the usage of the relative low threshold for online samples, resulting in overly conservative training and suboptimal performance. It is also supported by the results in Table 4. In summary, the empirical results confirm that the proposed dual thresholds are crucial for ensuring efficient performance improvement and stable training. The ablation study conclusively establishes dual thresholds mechanism as an indispensable component for effective online fine-tuning.

Ablation Study on Value-based Correction

SPA employs the value-based correction to revise the state proficiency determination of samples. To investigate the impact of value-based correction on policy training and validate its design rationality, we introduce two variants:

- **SPA-SA** implements simplified proficiency assessment by removing the classification correction.
- **SPA-RC** is a randomized correction variant where samples initially classified as high proficiency are randomly reclassified as either high proficiency or low proficiency during the classification correction phase.

Given that the correction mechanism primarily adjusts misclassified high proficiency samples to low proficiency ones, SPA-SA exhibits a lower overall policy constraint ratio compared to SPA. For SPA-RC, to ensure controlled comparison, we maintain the corrected samples proportion in SPA-RC equivalent to that in SPA. This configuration ensures that the overall policy constraint ratio remains the same in both algorithms. Such an experimental setup offers two key analytical advantages:

- The comparison between SPA-SA and SPA provides a comparative investigation of constraint ratio reduction effects.
- The SPA-RC vs. SPA comparison enables us to validate that the performance improvements stem primarily from accurate classification correction, rather than the aggressive learning derived from relaxed constraint.

As demonstrated in Figure 4, SPA achieves optimal performance compared to both ablation variants. Notably, SPA-RC and SPA-SA exhibit similar performance in Figure 4b, indicating that mere constraint ratio reduction through relaxed constraints fails to generate meaningful performance improvements.

	SPA	SPA-SA	SPA-RC
Precision	**60.22**	51.70	48.15
Recall	51.85	**81.72**	31.71

Table 5: The Precision (%) and Recall (%) of proficiency assessment on walker2d-medium-replay.

To further elucidate the fundamental mechanisms behind the effectiveness of value-based correction, we conduct an analysis on the proficiency assessment accuracy. From both offline and online datasets, we randomly select a subset of samples for accuracy evaluation. Using Monte Carlo-derived true Q-values from the offline and online data subsets, we conduct state proficiency assessment to establish ground truth state proficiency classification labels. Subsequently, we compare the classification results of various methods against the ground truth to evaluate the proficiency assessment accuracy. We adopt precision and recall, metrics commonly used in supervised learning (Sokolova and Lapalme 2009; Powers 2020) for accuracy evaluation. Given the imbalance in the quantities of high and low proficiency samples, these metrics are calculated specifically for high proficiency samples. The results in Table 5 indicate two crucial insights:

- SPA-SA exhibits recall-precision discrepancy, with higher recall than precision. This discrepancy can be attributed to the overestimate and underestimate discussed above. Specifically, misclassification inflates the high proficiency sample proportion, leading to a high recall paired with low precision. It clearly supports the motivation for classification correction and validates the necessity of the proposed value-based correction.
- SPA demonstrates significant precision improvements over SPA-SA, confirming the effectiveness of value-based correction in filtering erroneous classifications. It corrects the misclassification and thereby improving the accuracy of proficiency assessment. In contrast, SPA-RC, despite with correction, exhibits degraded precision and unacceptable recall reduction, further substantiating the superiority of proposed value-based correction in SPA.

In summary, these comprehensive results establish that the performance superiority of SPA originates from its effective value-based correction mechanism, which enables accurate and reasonable state proficiency assessment. It clearly indicates that accurate classification correction, rather than constraint ratio adjustments, drives the effectiveness in policy optimization.

Conclusion

This paper proposes the notion of state proficiency, a critical feature capturing the effectiveness of policy learning in a given state. Motivated by the significant variations in state proficiency across states, we propose SPA , which conducts adaptive fine-tuning based on state proficiency to achieve effective fine-tuning. Extensive empirical studies demonstrate that SPA significantly outperforms prior algorithms, confirming substantial improvements.

Acknowledgments

This work was supported by the Natural Science Research Foundation of Jilin Province of China under Grant No. 20220101106JC, and the National Natural Science Foundation of China under Grant No. 62407010.

References

Diehl, C.; Sebastian Sievernich, T.; Krüger, F., Hoffmann; and Bertram, T. 2023. Uncertainty-aware Model-based Offline Reinforcement Learning for automated driving. *IEEE Robotics and Automation Letters*, 1167–1174.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. In *Annual Conference on Neural Information Processing Systems*, 20132–20145.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *International Conference on Machine Learning*, 1587–1596.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*, 1856–1865.

Kong, R.; Wu, C.; Gao, C.-X.; Zhang, Z.; and Li, M. 2024. Efficient and Stable Offline-to-online Reinforcement Learning via Continual Policy Revitalization. In *International Joint Conference on Artificial Intelligence*, 4317–4325.

Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-learning for offline reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, 1179–1191.

Lee, S.; Seo, Y.; Lee, K.; Abbeel, P.; and Shin, J. 2021. Offline-to-Online Reinforcement Learning via Balanced Replay and Pessimistic Q-Ensemble. In *Conference on Robot Learning*, 1702–1712.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2022. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Luo, Q.-W.; Xie, M.-K.; Wang, Y.-W.; and Huang, S.-J. 2024. Optimistic Critic Reconstruction and Constrained Fine-Tuning for General Offline-to-Online RL. In *Annual Conference on Neural Information Processing Systems*, 108167–108207.

Lyu, J.; Ma, X.; Li, X.; and Lu, Z. 2022. Mildly conservative q-learning for offline reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, 1711–1724.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11): 2579–2605.

Nair, A.; Gupta, A.; Dalal, M.; and Levine, S. 2020. AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. *arXiv preprint arXiv:2006.09359*.

Nakamoto, M.; Zhai, S.; Singh, A.; Sobol Mark, M.; Ma, Y.; Finn, C.; Kumar, A.; and Levine, S. 2023. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. In *Annual Conference on Neural Information Processing Systems*, 62244–62269.

Powers, D. M. 2020. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Sokolova, M.; and Lapalme, G. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4): 427–437.

Sutton, R. S.; and Barto, A. G., eds. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Tang, S.; Makar, M.; Sjoding, M.; Doshi-Velez, F.; and Wiens, J. 2022. Leveraging Factored Action Spaces for Efficient Offline Reinforcement Larning in Healthcare. In *Annual Conference on Neural Information Processing Systems*, 34272–34286.

Wang, S.; Yang, Q.; Gao, J.; Lin, M.; Chen, H.; Wu, L.; Jia, N.; Song, S.; and Huang, G. 2023. Train once, get a family: State-adaptive balances for offline-to-online reinforcement learning. In *Annual Conference on Neural Information Processing Systems*.

Wu, J.; Wu, H.; Qiu, Z.; Wang, J.; and Long, M. 2022. Supported policy optimization for offline reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, 31278–31291.

Zhang, H.; Xu, W.; and Yu, H. 2023. Policy Expansion for Bridging Offline-to-Online Reinforcement Learning. In *International Conference on Learning Representations*.

Zhang, Y.; Liu, J.; Li, C.; Niu, Y.; Yang, Y.; Liu, Y.; and Ouyang, W. 2024. A Perspective of Q-value Estimation on Offline-to-Online Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 16908–16916.

Zhao, K.; Hao, J.; Ma, Y.; Liu, J.; Zheng, Y.; and Meng, Z. 2024. ENOTO: Improving Offline-to-Online Reinforcement Learning with Q-Ensembles. In *International Joint Conference on Artificial Intelligence*, 2609–2611.

Zhao, Y.; Boney, R.; Ilin, A.; Kannala, J.; and Pajarinen, J. 2022. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Zheng, H.; Luo, X.; Wei, P.; Song, X.; Li, D.; and Jiang, J. 2023. Adaptive policy learning for offline-to-online reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 11372–11380.