SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2024, Vol. 67, Iss. 7, 172203:1–172203:16
https://doi.org/10.1007/s11432-023-4028-1

Ensemble successor representations for task generalization in offline-to-online reinforcement learning

Changhong WANG[1], Xudong YU[1*], Chenjia BAI[2,4],
Qiaosheng ZHANG[2] & Zhen WANG[3*]

[1]*Space Control and Inertial Technology Research Center, Harbin Institute of Technology, Harbin* 150001*, China;*
[2]*Shanghai Artificial Intelligence Laboratory, Shanghai* 200232*, China;*
[3]*School of Cybersecurity, Northwestern Polytechnical University, Xi'an* 710072*, China;*
[4]*Shenzhen Research Institute of Northwestern Polytechnical University, Shenzhen* 518057*, China*

Received 23 July 2023/Revised 9 November 2023/Accepted 26 December 2023/Published online 25 June 2024

Abstract In reinforcement learning (RL), training a policy from scratch with online experiences can be inefficient because of the difficulties in exploration. Recently, offline RL provides a promising solution by giving an initialized offline policy, which can be refined through online interactions. However, existing approaches primarily perform offline and online learning in the same task, without considering the task generalization problem in offline-to-online adaptation. In real-world applications, it is common that we only have an offline dataset from a specific task while aiming for fast online-adaptation for several tasks. To address this problem, our work builds upon the investigation of successor representations for task generalization in online RL and extends the framework to incorporate offline-to-online learning. We demonstrate that the conventional paradigm using successor features cannot effectively utilize offline data and improve the performance for the new task by online fine-tuning. To mitigate this, we introduce a novel methodology that leverages offline data to acquire an ensemble of successor representations and subsequently constructs ensemble Q functions. This approach enables robust representation learning from datasets with different coverage and facilitates fast adaption of Q functions towards new tasks during the online fine-tuning phase. Extensive empirical evaluations provide compelling evidence showcasing the superior performance of our method in generalizing to diverse or even unseen tasks.

Keywords offline reinforcement learning, online fine-tuning, task generalization, successor representations, ensembles

1 Introduction

Reinforcement learning (RL) has emerged as a powerful approach for tackling complex sequential decision-making problems in various domains, such as games [1], robotics [2], manipulation [3], and autonomous driving [4,5]. However, most applications rely on extensive online interactions with the real environment or high-fidelity simulators, which can be infeasible or cost-expensive in real-world scenarios. Recently, offline RL [6] provides a promising solution to address this problem by learning an offline policy from a fixed dataset, without requiring online interactions. After that, the learned offline policy provides a good initialization for subsequent online fine-tuning. This paradigm allows efficient utilization of pre-collected data to provide an initialized policy and further improves the policy via limited online interactions [7,8].

Despite the potential of the offline-to-online paradigm to improve the sample efficiency in decision-making problems, existing approaches [8–11] are limited to the setting where the tasks of offline pre-training and online fine-tuning remain the same. In real-world applications, it is common to possess an offline dataset from a specific task while desiring policy generalization across various tasks. In such scenarios, the main challenge lies in effectively leveraging the information embedded within the offline datasets to benefit new tasks. For model-free methods, the policies and value functions are typically

* Corresponding author (email: hit20byu@gmail.com, zhenwang0@gmail.com)

learned for a specific task, thus fine-tuning them may cause poor performance on the new task due to overfitting on the offline datasets. For model-based methods, while their transition functions are invariant for different tasks, they necessitate accurate models for planning in downstream tasks and an exploratory dataset that covers the entire state-action space. Consequently, new approaches are required to facilitate generalization to new tasks during the fine-tuning stage while making full use of offline data.

Our study focuses on the task-generalization problem in an offline-to-online setting. In our work, we first explore the successor representation [12,13], which is a reward-agnostic representation that implicitly captures the underlying dynamics to predict future outcomes. By decoupling the dynamics from the reward function, successor representations enable rapid adaptation of the value function to novel tasks. Despite this advantage, we reveal that vanilla methods based on successor representation cannot learn effective policies during the fine-tuning phase. Moreover, we observe that the coverage of the offline data significantly impacts performance in offline-to-online generalization. Specifically, representations learned from the offline data with a narrow distribution is hard to generalize to novel tasks. Therefore, it is important to design an algorithm capable of conducting policy adaptation in new tasks with limited online interactions, while being robust to the data distributions of offline datasets.

To this end, we propose a novel approach that combines ensemble networks with successor representations to perform offline-to-online adaptation (ESR-O2O) in various downstream tasks. ESR-O2O adopts ensemble architecture to enhance the diversity of successor representations and value functions. This alleviates the dependency on data coverage or behavior policies during the offline training stage, allowing for learning useful and transferable representations even in scenarios where the available offline data exhibit a narrow distribution. During the online-adaptation stage, we keep the successor representations fixed and update task-specific parameters to learn value functions and policies for the downstream tasks. Theoretically, we establish that the optimality gap during online fine-tuning is bounded. Through empirical evaluations and comparisons, we show that ESR-O2O significantly outperforms existing RL methods for task generalization in offline-to-online settings.

Overall, our contributions are threefold. (1) To the best of our knowledge, our work is the first to investigate the task generalization problem in the context of offline-to-online RL, specifically focusing on pre-trained agents derived from a single offline environment. By addressing the reward gap, our work provides valuable insights into bridging the gap between offline pre-training and generalization in RL. (2) We introduce a novel approach that leverages ensembles to learn successor representations from datasets, thereby enhancing the robustness of the learned representations, especially when confronted with narrow offline distributions. This ensemble-based framework mitigates the dependence on data coverage and enhances the adaptability across various RL tasks, leading to improved performance in online fine-tuning. (3) We provide a substantial body of empirical results and a theoretical analysis to validate the feasibility and effectiveness of our method. We demonstrate that ESR-O2O effectively handles reward changes during fine-tuning, surpassing alternative approaches in terms of performance and sample efficiency.

2 Preliminaries

2.1 Reinforcement learning

We adopt the episodic Markov decision process (MDP) framework to formulate the sequential decision-making problem [14, 15]. Specifically, we define $M = (\mathcal{S}, \mathcal{A}, R, \mathcal{P}, \gamma, T)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, respectively. The transition dynamics of the environment is captured by the function $\mathcal{P}$, while R denotes the reward function. The discount factor γ represents the agent's preference for immediate rewards versus future rewards, and T specifies the length of an episode.

In online RL [16], the agent first observes the current state $s \in \mathcal{S}$ of the environment and applies an action $a \in \mathcal{A}$ to the environment based on its policy $\pi(a|s)$. The environment then gives the next state based on $\mathcal{P}$ and provides the agent with a scalar reward r. The agent repeats this process over time, with the goal of maximizing the expected cumulative reward, defined as $\mathbb{E}[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)]$. To quantify the value of a policy π, we define the state-action value function as follows:

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)|s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t), a_t \sim \pi(\cdot|s_t)\right]. \tag{1}$$

This function represents the expected cumulative reward starting from state s and taking action a under policy π. It can be updated by minimizing the temporal difference (TD) error $\mathbb{E}[(Q-\mathcal{T}Q)^2]$, where $\mathcal{T}Q = r+\gamma\mathbb{E}[\max_{a'}\bar{Q}(s',a')]$ is the Bellman operator and $\bar{Q}$ indicates the target network.

In offline RL, given a batch dataset D consisting of tuples (s,a,r,s'), where $r\sim R(\cdot|s,a)$ and $s'\sim \mathcal{P}(\cdot|s,a)$, agents are expected to find a policy $\hat{\pi}$ based on D so as to minimize the expected sub-optimality with respect to the optimal policy π^*, i.e., $\mathbb{E}_D[J(\pi^*)-J(\hat{\pi})]$, where the expectation is taken with respect to the randomness in the dataset. However, the bootstrapping process leads to an overestimation of the value for out-of-distribution (OOD) actions, because the bootstrapped error cannot be corrected without online interactions. Due to this, off-policy algorithms fail to learn useful policies from the static dataset. To overcome the bootstrapped error, several offline RL methods apply policy constraints [17–19] or conservative regularization to values [20], but may cause over-conservative estimation. There is another line of studies [21–24] that utilize ensembles for value functions to capture the epistemic uncertainty and attain favorable performance in many tasks. Specifically, the TD error in these methods becomes $\mathbb{E}[(\mathcal{T}Q-Q_i(s,a))^2]$, where the TD target could be shared for each ensemble network [21], i.e., $\mathcal{T}Q=r+\gamma\min\bar{Q}_i(s,a)$ or independently learned [22], i.e., $\mathcal{T}Q_i=r+\bar{Q}_i(s,a)$.

2.2 Successor representation

As an appealing approach to task transfer, successor representations (SR) [25], especially successor features (SF) [12] for continuous state space, have been proposed as a generalization for the value function. Let $M(s_t,a_t,s')=\sum_{i=0}^{\infty}\gamma^i p(s_{t+i}=s'|s_t,a_t)$ be the successor representation, defined as the discounted occupancy of state s', averaged over trajectories initiated in state s; then the state-action value function can be expressed as

$$Q(s,a)=\sum_{s'}M(s,a,s')R(s'). \tag{2}$$

The SR allows the decoupling of the dynamics of an MDP from its reward functions. The SR can intuitively be thought of as a predictive map that encodes each state in terms of the other states that will be visited in the near future. The SR can be learned in a similar way to temporal difference learning:

$$\delta_t(s')=\mathbb{I}[s_t=s']+\gamma M(s_{t+1},s')-M(s_t,s'), \tag{3}$$

where the error is the discrepancy between observed and expected state occupancy. The expected occupancy for states that are visited more frequently than expected should be increased, whereas the expected occupancy for states that are visited less frequently than expected should be decreased.

The SF extends SR to handle continuous state spaces by assuming that the reward function can be expressed as a linear combination of features ϕ and a weight vector w:

$$r(s_t,a_t,s_{t+1})=\phi(s_t,a_t,s_{t+1})^{\mathrm{T}}w. \tag{4}$$

Given the above decomposition, the Q function can be rewritten by

$$Q(s,a)=\mathbb{E}\left[\sum_{i=0}^{\infty}\gamma^i r(s_i,a_i)\right]=\mathbb{E}\left[\sum_{i=0}^{\infty}\gamma^i\phi^{\mathrm{T}}w\right]=\psi(s,a)^{\mathrm{T}}w. \tag{5}$$

In practice, the features ψ can be learned in many ways such as regression, using transition models or auto-encoders. The successor features ψ can also be learned by Bellman backups:

$$\bar{\psi}(s,a)=\phi(s,a)+\gamma\psi(s_{t+1},a'),\quad a'\sim\pi(\cdot|s_{t+1}). \tag{6}$$

The above formulation provides the possibility of quickly evaluating a policy π, thus being a promising way to handle the reward gap problem. When the reward function changes, we can keep the learned successor representation and re-learn the weight vector w by regression to obtain a new value function. However, the linear decomposition of the reward function is limited. Prior work addresses the reward generalization problem by considering a linear decomposition of the reward function and the value function. This modeling is restrictive and may fall short in capturing the complexity of the changed environment.

2.3 Problem setting

In this paper, we focus on the offline-to-online setting, where an agent is first pre-trained with offline datasets and then fine-tuned by interacting with the environment. Several studies [7, 26] found that the agent exhibits a huge dip at the initial stage of online fine-tuning, which is not expected in real applications. More importantly, we put attention to the reward generalization problem that the pre-trained agents are fine-tuned in a new environment with the same dynamics but different reward functions.

Here, we present a clear formulation for our problem setting. Let $M_i = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R_i, \gamma)$ be the MDP in the offline training stage and logged data are collected following the reward function R_i. Then we introduce a different MDP as $M_j = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R_j, \gamma)$ for online fine-tuning of the policy after pertaining in M_i. We remark that M_i and M_j are different tasks with the same dynamics but different reward functions. In the offline training stage, agents learn representations, value functions, and policies from offline data. To perform policy generalization in the fine-tuning stage with a different task M_j, a robust and favorable performance is expected to provide a good initialization and fast adaptation.

3 Related work

Offline-to-online RL. Our work contributes to offline-to-online RL by fine-tuning pre-trained offline agents through online interactions. Previous studies have primarily addressed distributional shift, which refers to the disparity between offline data and online transitions. Various approaches have been proposed to mitigate the performance drop during the initial fine-tuning stage caused by distributional shift. These approaches include adaptive adjustment of behavior cloning weights [27], reconstruction of the replay buffer and sampling methods [8], modification of the update target of the critic network [7, 9, 10], and adaptive policy composition [11]. Recently, E2O [28] and PROTO [29] have achieved significant performance by integrating offline pessimism and online optimism with ensembles, or using an iteratively evolving regularization term and performing a trust-region-style update. While both E2O and our approach utilize ensembles, their goals and solved problems differ. E2O employs Q ensembles to perform pessimism to prevent over-estimation during offline training and to encourage exploration during online fine-tuning. However, our approach uses ensembles to model diverse patterns of successor representations and Q functions, allowing robust representation learning despite offline data limitations. More importantly, these offline-to-online algorithms mainly address scenarios with consistent offline and online environments. The issue of distributional shift and generalization across different tasks has received limited attention. To the best of our knowledge, our work is the first to investigate the problem of reward generalization in the offline-to-online setting.

Task generalization. Our work aims to deal with the task generalization problem. While recent studies [30–33] have made progress in addressing generalization and distributional shift challenges, they focus on specific aspects of the offline-to-online learning process. These studies either consider minor changes in dynamics [30], emphasize generalization in an online setting [33], or primarily focus on offline learning [32], which is constrained by dataset quality. On the other hand, successor features are utilized to handle learning in dissimilar environments [34, 35]. But these approaches consider this problem in an online setting and assume the availability of an effective exploration policy during the learning process. In contrast, our work focuses on applying pre-trained agents, including learned representations, value functions, and policies, to a different online environment. Furthermore, many existing methods that utilize successor representations also assume the use of a set of training tasks, while our work does not impose such a limitation. By focusing on the reward generalization problem and the offline-to-online setting, our work aims to bridge the gap between offline pre-training and online fine-tuning, contributing to a deeper understanding of generalization in reinforcement learning.

Successor representation. Our method leverages successor representation to capture the environmental dynamics, enabling generalization across different reward functions. Successor representations serve as predictive representations that summarize the successive features to follow. They also provide mechanic explanations similar to human understanding [36]. Several previous studies have combined successor representation with Generalized Policy Improvement [13] to transfer behaviors across navigation tasks [34]. Recently, the unsupervised representation learning paradigm combined with the successor measure is discussed [37,38], avoiding learning basic features for conventional successor features. Nevertheless, these studies often rely on effective exploration policies or exclusively diverse datasets. In contrast, our

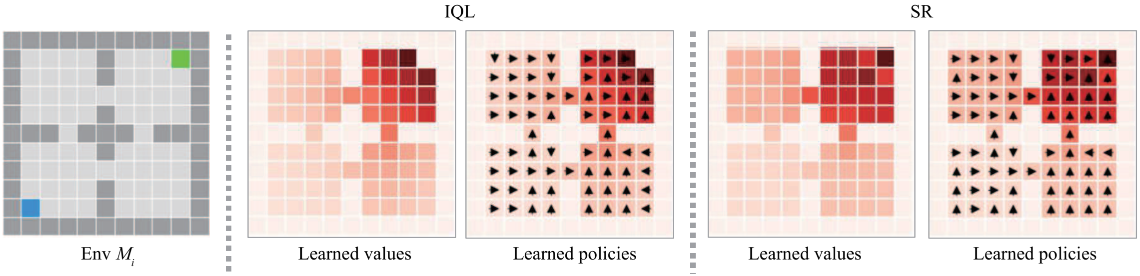

Figure 1 (Color online) Offline pre-training in a four-room maze navigation task. In the left picture, the blue block indicates the starting point, and the green one indicates the goal. Two middle pictures describe the values and policies learned by IQL from offline data, where the arrow represents the action to be taken at each position. The two right pictures show the values and policies learned by vanilla SR-based methods. After offline training, both IQL and SR-based methods can learn correct values and the optimal policy.

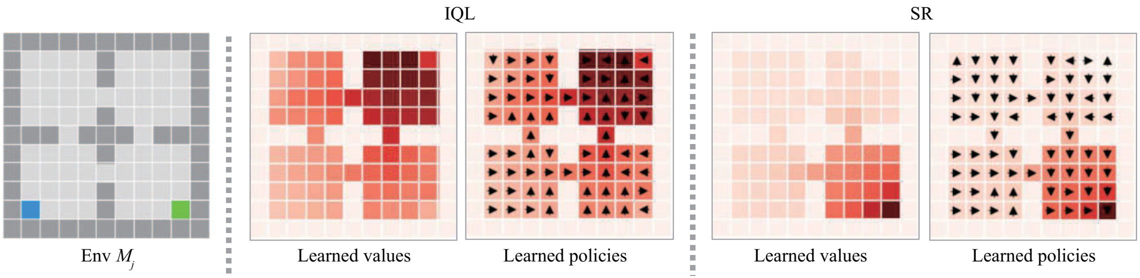

Figure 2 (Color online) Online fine-tuning in a navigation task with a different goal. When the goal changes, the reward function changes accordingly. In this setting, IQL fails to learn correct values and policies in the new environment, while SR-based methods re-learn the values and lead to effective policies.

work analyzes learning representations from various types of datasets, expanding the applicability and potential of successor representations.

Ensemble. Our work is also related to ensemble-based methods. In model-free RL, ensemble methods have gained considerable interest for estimating epistemic uncertainty for action-value estimates. In online-RL, ensembles are frequently employed to enhance exploration [39, 40] by encouraging agents to seek out actions with the highest variance in estimated values. This is achieved by constructing a distribution of action-value estimates using the ensemble and acting optimistically with respect to the upper bound [41–44]. In offline-RL, ensembles of Q functions or environmental models are used to obtain conservation estimation from the dataset [21, 22, 24, 45]. Deep ensembles [46] are shown to effectively capture epistemic uncertainty arising from incomplete information and approximate the true posterior distribution. By leveraging ensembles, the diversity of networks is enhanced, thereby mitigating estimation bias for the value function [47]. Inspired by this, our work adopts ensemble networks to improve the estimation of successor representations and value functions.

4 Motivating examples

In this section, we provide motivating examples to illustrate the challenges in offline-to-online RL, especially when dealing with reward gaps. We also highlight the limitations of vanilla successor representation-based methods when learning from offline data. First, we use a tabular case to show that existing offline-to-online RL approaches struggle with reward generalization. In contrast, successor representations can enhance adaptive value function learning. Furthermore, we explore the limitation of vanilla SR-based methods in task generalization after pre-training on a single offline environment. Specifically, we find that the coverage of the offline data affects both pre-trained performance and online fine-tuning performance.

Grid world example. Consider a navigation task in a four-room environment, with each room divided into a 4×4 grid of cells, as depicted in Figures 1 and 2. In this scenario, the blue block marks the starting point, and the green block indicates the goal. Our paper focuses on a specific setting where the agent has access to a dataset from the environment M_i. The objective is to apply pre-trained policies from the offline dataset to a new environment M_j. In the grid world example, M_i and M_j possess different reward functions, achieved by altering the goal's location while keeping the transition dynamics constant.

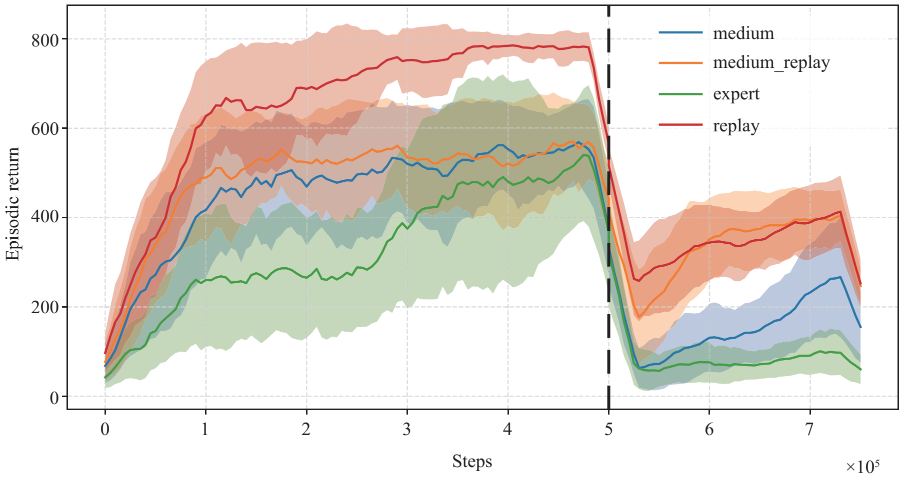

Figure 3 (Color online) Pre-training and fine-tuning performance of vanilla SR-based methods under different data distributions. The black dashed line indicates the transition from offline pre-training to online fine-tuning when the task changes. The shaded area represents the variance of performance across multiple experiments conducted with 5 random seeds. Representations learned from more diverse datasets, such as 'medium-replay' and 'replay' data, exhibit superior performance compared to representations learned from narrower datasets ('medium', 'expert').

To illustrate the challenge of task generalization in offline-to-online RL, we evaluate two methods: the state-of-the-art algorithm IQL and a vanilla successor representation-based method. In Figure 2, IQL successfully learns the optimal values in the original environment M_i during the pre-training phase. However, when attempting to fine-tune the learned values in the new environment M_j, IQL fails to generate accurate value estimations. This limitation arises from the agent's specialization in training for specific reward functions, hindering its ability to generalize effectively to novel reward functions. In contrast, values learned by the vanilla SR-based method show adaptability to new environments with just a few fine-tuning steps. This comparison highlights the limitations of current offline-to-online RL methods and the potential of SR-based methods in addressing the task generalization problem.

The impact of offline data. Despite the advantage of adapting to new tasks, challenges persist in learning from offline data using vanilla SR-based methods (as presented in Subsection 2.2). Here, we consider a more complex scenario with continuous state and action space, where representations and values are expressed by neural networks. Specifically, we test a quadruped robot's ability to fine-tune its performance from a rolling task to a walking task after training on an offline dataset. This transition introduces a reward gap, presenting an offline-to-online RL challenge. To investigate this, we use four types of offline datasets with varying coverage: 'medium' data from a medium-level agent, 'medium-replay' data encompassing all experiences in training a medium-level agent, 'expert' data generated by an expert-level agent, and 'replay' data including all experiences in training an expert-level agent. Our approach involves the initial extraction of successor representations from the offline data, followed by policy and value function fine-tuning based on online interactions.

Address value overestimation during offline pre-training due to distributional shift is crucial. In vanilla SR-based methods, both representations and values are updated via bootstrapping. The bootstrapping process can result in overestimation [17], since the bootstrapped error cannot be corrected during offline learning. To mitigate this, we make two key modifications. We set the minimum of two critic estimates as the temporal difference (TD) target [48] and employ layer normalization to prevent catastrophic overestimation [23]. As shown in Figure 3, these improvements help the vanilla SR-based method to learn policies from offline data. During the pre-training stage, we observe that the performance is affected by the coverage and quality of offline dataset. For example, agents trained on 'replay' data tend to exhibit better performance compared to those trained on 'expert' data.

In addition to the overestimation problem, another challenge is the reward gap, which exacerbates the discrepancy between offline datasets and online transitions. During fine-tuning, we find that agents pre-trained on 'replay' or 'medium-replay' data outperform those trained on 'expert' or 'medium' data. This highlights the significant influence of data coverage on successor representation learning and subsequent fine-tuning performance. In other words, greater diversity in offline data leads to more effective learned representations and improved performance. However, practical concerns arise as there are no guarantees regarding the diversity and quality of offline datasets. Hence, the challenge remains in learning valuable successor representations from datasets with different coverage and quality.

5 Methodology

In this section, we present our approaches to address task generalization challenges in offline-to-online RL. First, we analyze the sub-optimality gap during the fine-tuning phase when a reward gap exists. Based on this analysis, we propose a simple yet effective method to improve the ability to generalize to unseen tasks by incorporating ensembles. This method diversifies successor representations and critic networks, reducing the bias in representations and value functions. We then outline the pre-training and fine-tuning process that leverages ensemble successor representations to tackle the task generalization problem.

5.1 Theoretical analysis

To figure out how to handle the task generalization problem, we delve into the theoretical foundations to characterize the sub-optimality gap in the fine-tuning phase. First of all, let us introduce some notations for clarity. Consider the offline environment M_i with reward function r_i and the online environment M_j with r_j. Let π_i^* and π_j^* denote the optimal policies in M_i and M_j, respectively. We introduce an optimal successor feature ψ^{π^*} by considering $Q^{\pi^*} = \psi^{\pi^*} w$. This optimal successor feature represents the future state occupancy by following policy π^*. In M_i and M_j, the optimal successor features can be expressed as $\psi^{\pi_i^*}$ and $\psi^{\pi_j^*}$. Their corresponding optimal value functions are defined as $Q_i^{\pi_i^*} = \psi^{\pi_i^*} w_i$ and $Q_j^{\pi_j^*} = \psi^{\pi_j^*} w_j$, where w_i and w_j indicate the weight vectors as shown in (4).

In our work, we pre-train agents from the offline dataset and then fine-tune them in the online phase. We define the pre-trained successor feature as $\hat{\psi}$ and pre-trained policies as $\hat{\pi}$. Then, we can get the following performance bound during the fine-tuning stage.

Proposition 1 (Sub-optimality gap). For all $s \in \mathcal{S}, a \in \mathcal{A}$, let the learned value function after the fine-tuning stage be $Q_j^{\pi} = \hat{\psi} w^{\pi}$; then the fine-tuning performance bound can be expressed as

$$|Q_j^{\pi_j^*} - Q_j^{\pi}| \leqslant \|w_j\|_\infty \|\psi^{\pi_j^*} - \hat{\psi}\|_1 + \|\hat{\psi}\|_\infty \|w_j - w^{\pi}\|_1. \tag{7}$$

Proof. The sub-optimality gap can be decomposed according to the triangle inequality and Holder's inequality:

$$\begin{aligned} |Q_j^{\pi_j^*} - Q_j^{\pi}| &= |Q_j^{\pi_j^*} - Q_j^{\hat{\pi}} + Q_j^{\hat{\pi}} - Q_j^{\pi}| \leqslant |Q_j^{\pi_j^*} - Q_j^{\hat{\pi}}| + |Q_j^{\hat{\pi}} - Q_j^{\pi}| \qquad (8) \\ &\leqslant \|w_j\|_\infty \|\psi^{\pi_j^*} - \hat{\psi}\|_1 + \|\hat{\psi}\|_\infty \|w_j - w^{\pi}\|_1. \qquad (9) \end{aligned}$$

This proposition indicates that the sub-optimality gap in the fine-tuning stage is influenced by the optimality of successor features and the weight vector. The term $\|\psi^{\pi_j^*} - \hat{\psi}\|_1$ represents the difference between the optimal successor feature in M_j and the pre-trained successor feature. Since the successor feature captures the dynamics information, this difference could be interpreted as the disparity between the dynamics captured by the offline dataset and the dynamics in M_j. The term $\|w_j - w^{\pi}\|_1$ quantifies the approximation error for the true reward function in M_j. If we retain the pre-trained value function from the offline dataset and assume that w^{π} approximates w_i well, then $w_j - w^{\pi}$ contains a transition from w_i to w_j. When the reward gap $\|w_i - w_j\|_1$ is large, this transition can be challenging due to the gradient propagation mechanism in neural networks. Consequently, the performance of the fine-tuning process can be compromised.

We further bound the two terms in (7) under several assumptions. Assuming that the norm of the basic features and the rewards are bounded, i.e., $\|\phi\|_2 \leqslant 1$ and $|r| \leqslant r_{\max}$, we can establish a similar inference process as in linear MDP. The following lemma demonstrates that the gaps in fine-tuning performance are bounded. Specifically, the reward approximation error depends on the quality and size of the data buffer collected during fine-tuning, while the representation gap is bounded by the estimated error of the dynamics.

Lemma 1 ([49, 50]). Let $Z = \sqrt{\lambda} + r_{\max} \cdot \sqrt{2\log\frac{1}{\delta} + d\log(1+\frac{N}{\lambda d})}$, $\Lambda = \lambda I + \sum_{i=1}^{N}\phi(s_i,a_i)\phi(s_i,a_i)^{\mathrm{T}}$; then the following inequality holds with probability $1-\delta$:

$$\|w_j - w^{\pi}\|_1 \leqslant \sqrt{d}\|w_j - w^{\pi}\|_{\Lambda} \leqslant \sqrt{d}Z, \tag{10}$$

where $w_j, w^{\pi} \in \mathbb{R}^d$, $\|w\|_{\Lambda} = \sqrt{w^{\mathrm{T}}\Lambda w}$, λ is the regularization parameter, and N denotes the size of the data buffer. For successor representations, we have

$$\|\psi^{\pi_j^*} - \hat{\psi}\|_1 \leqslant \frac{\gamma}{(1-\gamma)^2}\|\mathcal{P}^* - \mathcal{P}\|_1, \tag{11}$$

where $\mathcal{P}^*$ and $\mathcal{P}$ denote the true and estimated dynamics, respectively.

Proof. For (10), we simplify the equation by setting $\bar{w} = w_j - w^{\pi}$; then we have

$$\|\bar{w}\|_1 \leqslant \sqrt{d}\|\bar{w}\|_2 \leqslant \sqrt{d}\sqrt{\|\bar{w}\|_2^2 + \|\phi^{\mathrm{T}}\bar{w}\|_2^2} = \sqrt{d}\sqrt{\bar{w}^{\mathrm{T}}(I+\phi^{\mathrm{T}}\phi)\bar{w}} = \sqrt{d}\|\bar{w}\|_{\Lambda} \leqslant \sqrt{d}Z. \tag{12}$$

Please refer to Theorem 16 in [49] and Theorem 2 in [50] for more detailed proof.

Based on this lemma, the estimation error of the successor representation and the weight vector is bounded. Since $\|\hat{\psi}\|_{\infty} = \|\sum_{i=0}^{\infty}\gamma^i\phi^{\mathrm{T}}\|_{\infty} = \|\frac{1}{1-\gamma}\phi^{\mathrm{T}}\|_{\infty} \leqslant \frac{1}{1-\gamma}$, the sub-optimality gap in Proposition 1 is upper bounded.

5.2 Randomized ensembles of successor representations and critic networks

In Section 4 and Subsection 5.1, we provide empirical and theoretical insights about learning successor representations from offline data and use them to deal with task generalization in the fine-tuning stage. Specifically, we observe that the fine-tuning performance is influenced by the learning process of value functions, represented by weight vectors, and the pre-trained representations. The quality of pre-trained representations depends on the quality and coverage of the offline data, as well as the handling of catastrophic overestimation. To this end, we expect agents to acquire well-performed representations regardless of the offline data distribution and demonstrate robust task generalization capabilities.

In this paper, we propose a novel approach utilizing randomized ensembles of successor representations and critic networks, named by ESR-O2O, to address the challenges discussed earlier. Our design is motivated by several reasons. First, multiple estimates enable characterization for epistemic uncertainty and the lower confidence bound of estimations for representations and values, facilitating efficient and pessimistic learning in the offline stage [51, 52]. Second, ensemble helps to mitigate the estimation bias [47], thus reducing the gaps in the sub-optimality bound described in (7). Third, ensembles are beneficial to enhancing the sample efficiency [23, 41]. Finally, ensembles impose diversity on the estimation, thereby mitigating the limitation of the coverage of offline datasets.

We compare our method ESR-O2O with vanilla SR-based methods and show the differences in the framework in Figure 4. Vanilla SR-base methods often assume access to multiple source environments to learn representations. In such cases, the generalization to target environments can be considered the interpolation [34] or modeling the Gaussian process [35]. However, our method faces the more challenging task of task generalization with access to only an offline dataset generated from a single environment. On the other hand, vanilla methods typically assume a linear relationship between the value function and the successor representation, which restricts the applicability of successor representations. In contrast, our method directly models the value function as a function of successor representations, utilizing multiple layers rather than a single linear layer. This broader modeling capability enhances the feasibility and applications of our method in more complex scenarios.

Formally, we introduce the ensemble successor representations and ensemble Q functions as follows. As depicted in Figure 4(b), $\psi_i : i \in [1,n], |\mathcal{S}| \times |\mathcal{A}| \to |\mathcal{S}|$ represents the ensemble successor representations, where each member is initialized randomly. Similarly, $Q_k : k \in [1,n], |\mathcal{S}| \times |\mathcal{A}| \to 1$ represents the ensemble Q functions. Both the ensemble SR and ensemble Q functions are updated using TD learning:

$$\psi_k(s_t,a_t) \leftarrow \psi_k(s_t,a_t) + \alpha[\phi(s_t,a_t) + \gamma \cdot \psi_k(s_{t+1},a_{t+1}) - \psi_k(s_t,a_t)], \tag{13}$$

$$Q_k(\psi_k(s_t,a_t)) \leftarrow Q_k(\psi_k(s_t,a_t)) + \alpha[r(s_t,a_t) + \gamma \cdot Q_k(\psi_k(s_{t+1},a_{t+1})) - Q_k(\psi_k(s_t,a_t))], \tag{14}$$

where α and γ represent the learning rate and discount factor, respectively. The basic feature $\psi(\cdot)$ is assumed to be available in this work. When the input is finite and vectorized, this mapping can be

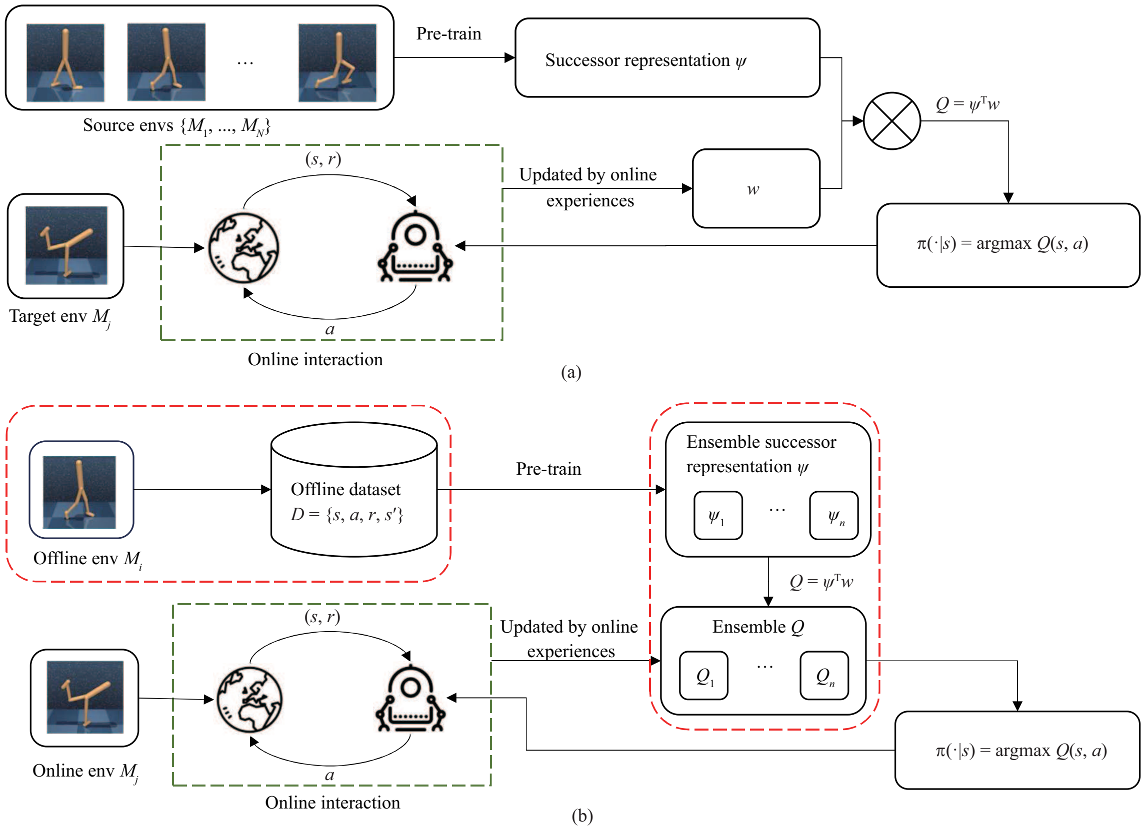

Figure 4 (Color online) Frameworks of (a) vanilla SR-based methods and (b) ESR-O2O for offline-to-online learning. Red dashed boxes indicate the different parts of ESR-O2O from vanilla SR. While vanilla SR trains multiple representations from source environments, ESR-O2O extracts ensemble representations using offline datasets from a single environment. Another difference lies in the construction of the Q function, where vanilla SR considers linear composition, and ESR-O2O incorporates ensemble Q functions. Other parts are the same, including fine-tuning the value function with online interactions with M_j, which follows a greedy policy $\pi(\cdot|s) = \arg\max Q(s, a)$.

Algorithm 1 Learning representation, policy, and value function from offline data

---

Require: Offline dataset D generated from M_i, policy network π, Q-networks $\{Q_k\}_{k\in[n]}$, and target Q-networks $\{\bar{Q}_k\}_{k\in[n]}$, representation networks $\{\psi_k\}_{k\in[n]}$, and target representation networks $\{\bar{\psi}_k\}_{k\in[n]}$.
1: Initialize the parameters of π, $\{Q_k\}_{k\in[n]}$, $\{\bar{Q}_k\}_{k\in[n]}$, $\{\psi_k\}_{k\in[n]}$, and $\{\bar{\psi}_k\}_{k\in[n]}$;
2: **while** Not coverage **do**
3: Sample transitions $\{(s, a, r, s')\}$ from D;
4: Update successor representations with (13);
5: Update critic networks using (14);
6: Update target networks using (15);
7: Update the policy network greedily according to (16);
8: **end while**

---

simplified as the identity function. The input of the Q functions is the successor representation, which differs from linear composition (i.e., $Q = \psi^{\mathrm{T}} w$). It is noted that we utilize independent targets instead of shared targets, which may introduce optimism in certain cases [22].

5.3 Offline pre-training and fine-tuning process

In this subsection, we present the offline pre-training process and online fine-tuning process. Algorithm 1 outlines our approach. During the offline training stage, the representations and the critic network are trained based on the Bellman equation. In each mini-batch, all networks in the ensemble are updated using (13) and (14). Target networks are used to stabilize the learning process. If we define the respective parameters of the original network and the target network as θ and $\bar{\theta}$, then target networks can be updated using Polyak averaging:

$$\bar{\theta} \leftarrow \rho\bar{\theta} + (1 - \rho)\theta. \tag{15}$$

Algorithm 2 Online fine-tuning process given pre-trained ESR

Require: Online environment M_j, fine-tuning steps T;

Load pre-trained policy network π, pre-trained critic networks Q_i^{π}, target networks $\bar{Q}_i^{\pi}$, and learned SR ψ^{π};

Fix the parameters of representation networks ψ^{π};

repeat

  Interact with the environment M_j with action $a \sim \pi(\cdot|s)$;

  Receive new state s' and reward r from the environment;

  Store transitions $\{s, a, r, s'\}$ into replay buffer;

  Update critic networks and policies according to (14) and (16);

  Update target networks;

until online steps $= T$.

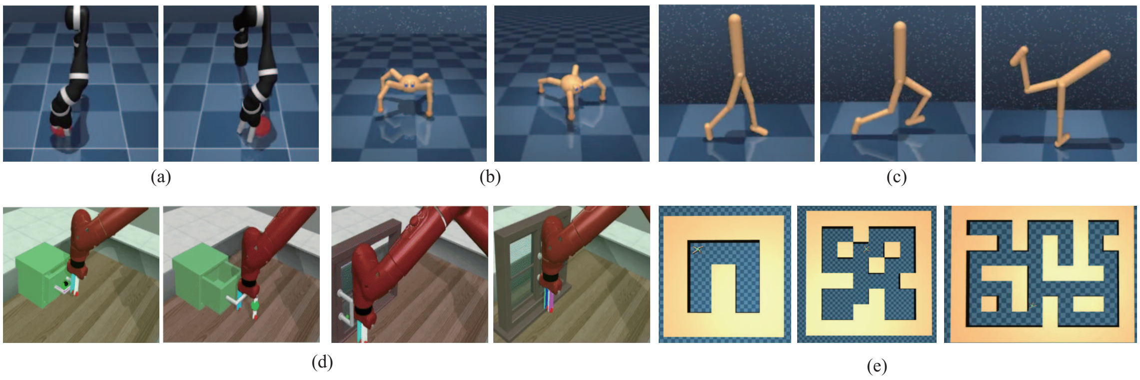

Figure 5 (Color online) Experimental environments, including (a) Reach from UTDS [53], (b) Quadruped, (c) Walker, (d) MetaWorld [54], and (e) Antmaze from D4RL [55]. These environments can be categorized into three classes based on the magnitude of the reward gap: small gap (Quadruped, Walker), big gap (Reach, MetaWorld), and no gap (Antmaze).

The policy network is trained to maximize the minimum value among the ensemble of critics, which can be formulated as follows:

$$\pi(\cdot|s) \leftarrow \arg\max_a \min_k Q_k(s,a) = \arg\max_a \min_k Q_k(\psi_k(s,a)). \tag{16}$$

This can be thought of as forming a lower confidence bound (LCB) for the value function of a policy using the batch data and then seeking to find a policy that maximizes the LCB.

When it comes to online fine-tuning, ESR-O2O loads all pre-trained networks, including the policy network, pre-trained critic networks, target networks, and the representation network. We do not set many fine-tuning steps, since a large number of online interactions are not available. To prevent the representation from being compromised or even destroyed [11], we fix the parameters of representation networks during the fine-tuning process. Algorithm 2 outlines our online fine-tuning procedure.

6 Experiments

In this section, we present experimental evaluations to assess the effectiveness and feasibility of our proposed method. Specifically, we aim to address the following research questions: (1) Does the utilization of ensembles for successor representations and Q functions lead to improved performance in both the offline learning and fine-tuning stages? (2) In scenarios without reward gaps, does ensemble SR outperform existing offline-to-online learning approaches? (3) What is the significance of ensembles in achieving superior performance with our method?

6.1 Setups

All experimental environments are illustrated in Figure 5 [53–55]. The Quadruped environment involves tasks such as walking, running, jumping, and rolling, while the Walker environment focuses on walking, running, and flipping tasks. In the Reach environment, a manipulator is required to place blocks in different positions, including bottom left, bottom right, top left, and top right. Each goal corresponds to a distinct reward function. To investigate task generalization during fine-tuning from a single offline environment, we randomly select two tasks, denoted by M_i and M_j, from the aforementioned environments. Agents pre-trained on the offline dataset generated in M_i are then fine-tuned and evaluated in

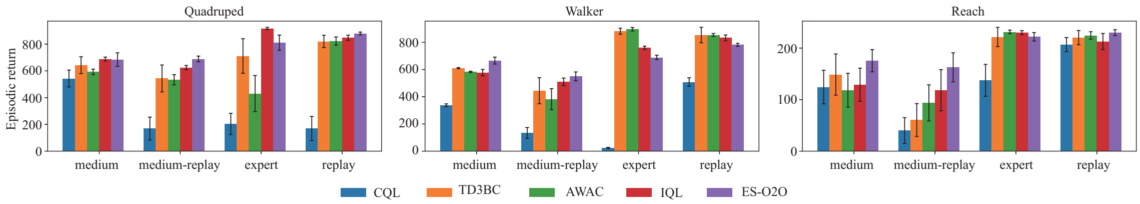

Figure 6 (Color online) Offline performance after 1 M training steps. Each bar represents the performance averaged on different tasks in the same environments. The error bars are estimated across 5 random seeds.

Table 1 Average return on MetaWorld tasks[a)]

Environment	PEX	PROTO	ESR-O2O
M_i: drawer-close-v2	3409.04 ± 1702.83	3646.20 ± 885.18	3249.58 ± 1874.60
M_j: drawer-open-v2	812.49 ± 789.24	395.71 ± 29.85	**2278.42** ± 275.47
M_i: window-open-v2	394.35 ± 255.25	977.18 ± 870.05	280.81 ± 15.8604
M_j: window-close-v2	571.73 ± 369.45	174.69 ± 164.64	**2910.27** ± 1446.12

a) The best results in online fine-tuning are in bold.

the new environment M_j, which shares the same dynamics but possesses a different reward function. We refer to Appendix A for more details.

6.2 Performance when reward functions change

Due to the absence of available baselines specific to our setting, we incorporate several offline-to-online learning methods like AWAC [7], Off2On [8], PEX [11], and PROTO [29] as baselines for comparison. We also compare our method with offline RL methods such as CQL [20], IQL [9], and TD3BC [19], and present their fine-tuning results in Appendix B due to page limitations. To assess the performance, we categorize the environments based on the magnitude of the reward gap between M_i and M_j, as shown in Figure 5. When the reward gap is small, previously learned policies and value functions may be effectively utilized, implying easier generalization. In contrast, harder generalization scenarios involve a large reward gap, making it challenging for the learned policies to perform well in the new task.

Offline performance. We begin by comparing the performance of pre-trained agents using various algorithms. As illustrated in Figure 6, our method outperforms the start-of-the-art offline RL algorithms with 'medium' and 'medium-replay' data, and exhibits competitive performance given 'expert' and 'replay' data. Furthermore, our method achieves reduced variances in episodic returns compared to baseline algorithms, showing enhanced stability. We also observe that the performance gaps across different datasets, such as the disparity between 'replay' data and 'medium-replay' data, are more pronounced in baseline methods, whereas our proposed method displays narrower performance gaps among these datasets. This implies that ESR-O2O is more robust to the data coverage and quality of offline datasets.

Fine-tuning performance with a big reward gap. In scenarios with a significant reward gap, previously learned policies or value functions become ineffective. As presented in Figure 7, conventional methods fail to acquire useful policies in the fine-tuning stage. However, ESR-O2O demonstrates the ability to effectively handle such reward gaps and achieve robust task generalization. The performance of ESR-O2O on 'medium-replay' data closely approaches that of 'replay' data, indicating its capacity to learn effectively even from datasets lacking expert policies. When 'expert' data is provided, ESR-O2O exhibits favorable performance during offline training, albeit the fine-tuning performance is inferior to those on other types of offline data. This suggests that in the presence of a highly narrow dataset, the ability of ESR-O2O to diversify representations and Q functions and improve performance through limited interactions may be constrained. PEX also demonstrates adaptability in several tasks due to its adaptive policy composition, but still falls short of our method's performance. We also present the comparisons in the MetaWorld benchmark in Table 1. These results also validate the superiority of our method when fine-tuning in a new environment, with more detailed results available in Appendix B.

Fine-tuning performance with a small reward gap. In scenarios with small reward gaps, it is possible to leverage pre-trained policies or value functions for the new task. PEX and AWAC demonstrate a degree of robustness in task switching. Conversely, Off2On, which is built on top of CQL and extracts policies by maximizing Q functions, appears to be more sensitive to the distributional shift, resulting

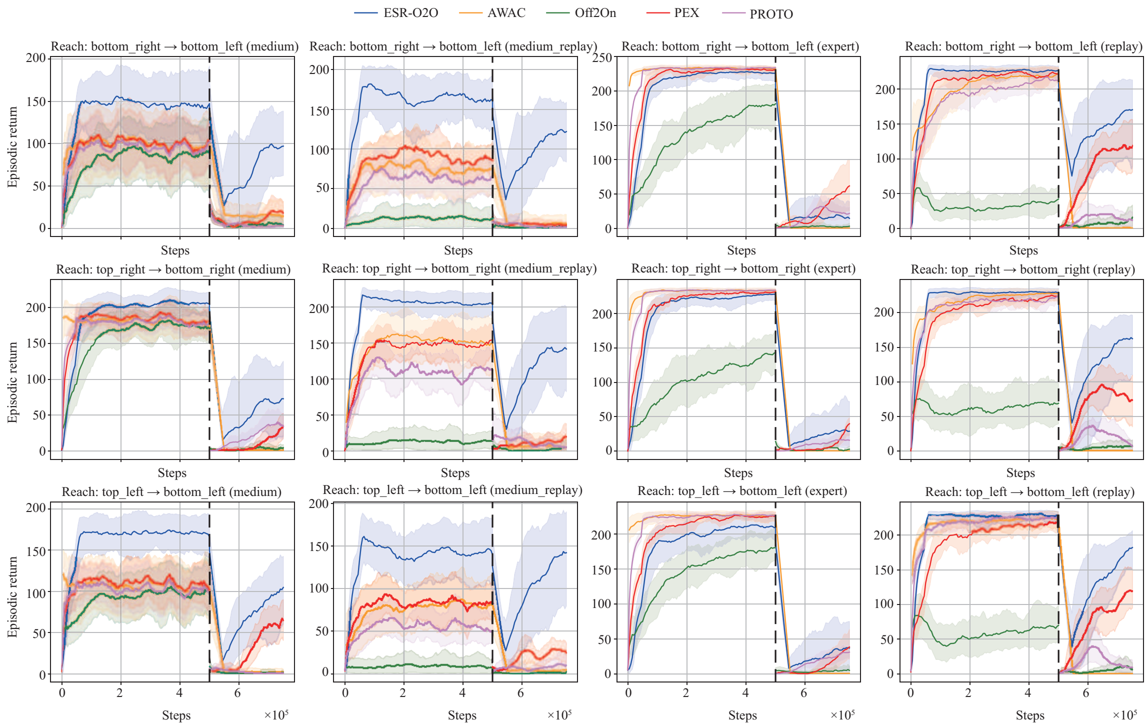

Figure 7 (Color online) Fine-tuning performance in scenarios with large reward gaps. While baseline methods struggle to perform well during the fine-tuning stage, ESR-O2O demonstrates significant improvements in the fine-tuned policies, regardless of the quality of the offline datasets.

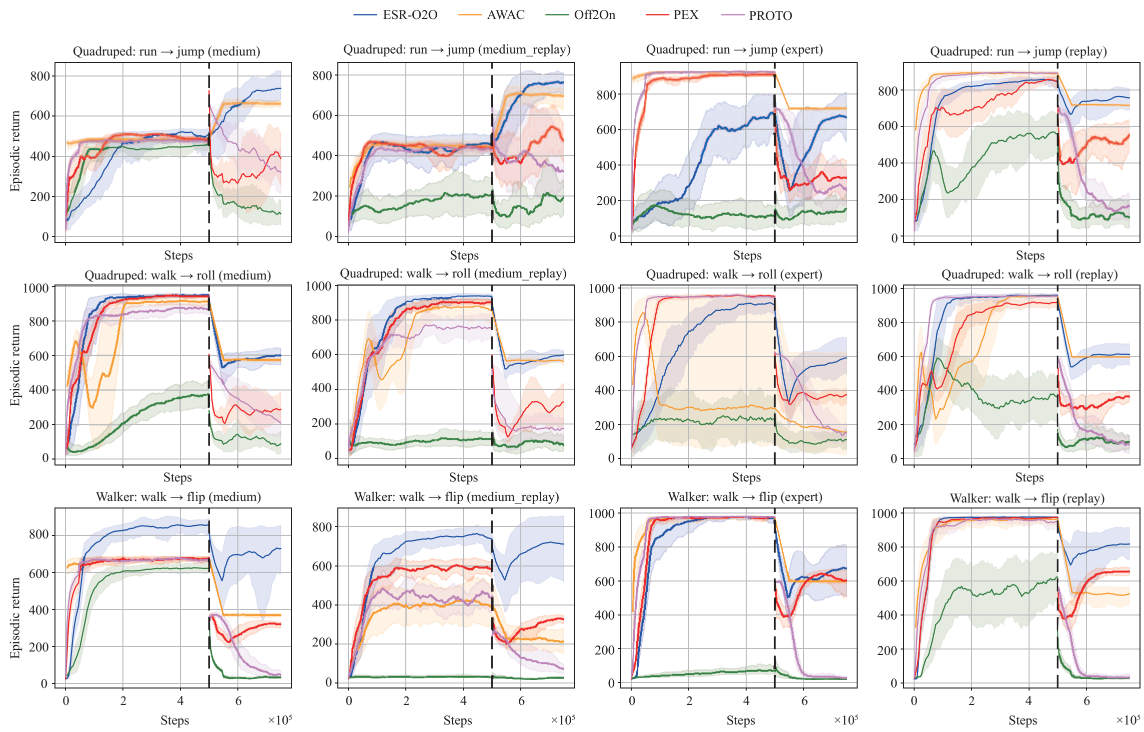

Figure 8 (Color online) Fine-tuning performance comparisons in scenarios with small reward gaps. The shaded areas indicate variances across 5 random seeds.

in instability. Figure 8 illustrates that ESR-O2O significantly outperforms other methods. While PEX and PROTO swiftly acquire useful policies during pre-training, we observe performance degradation and inferiority during fine-tuning in a new environment. We speculate this is due to the lack of consideration

Table 2 Performance on Antmaze tasks[a)]

Environment	AWAC	CQL	IQL
antmaze-umaze-v2	52.75 ±8.67 → 98.75 ±1.09	94.00 ±1.58 → 99.50 ±0.87	77.00 ±0.71 → 96.50 ±1.12
antmaze-umaze-diverse-v2	56.00 ±2.74 → 0.00 ±0.00	9.50 ±9.91 → **99.00 ± 1.22**	59.50 ±9.55 → 63.75 ±25.02
antmaze-medium-play-v2	0.00 ±0.00 → 0.00 ±0.00	59.00 ±11.18 → 97.75 ±1.30	71.75 ±2.95 → 89.75 ±1.09
antmaze-medium-diverse-v2	0.00 ±0.00 → 0.00 ±0.00	63.50 ±6.84 → 97.25 ±1.92	**64.25 ± 1.92** → 92.25 ±2.86
antmaze-large-play-v2	0.00 ±0.00 → 0.00 ±0.00	28.75 ±7.76 → 88.25 ±2.28	38.50 ±8.73 → 64.50 ±17.04
antmaze-large-diverse-v2	0.00 ±0.00 → 0.00 ±0.00	35.50 ±3.64 → 91.75 ±3.96	26.75 ±3.77 → 64.25 ±4.15
Average	18.12 → 16.46	48.38 → 95.58	56.29 → 78.50
Environment	**Cal-QL**	**ESR-O2O**	
antmaze-umaze-v2	76.75 ± 7.53 → **99.75 ± 0.43**	**98.00 ± 1.87** → 99.2 ± 1.79	
antmaze-umaze-diverse-v2	32.00 ± 27.79 → 98.50 ± 1.12	**93.75 ± 2.63** → 98.75 ± 0.96	
antmaze-medium-play-v2	71.75 ± 3.27 → **98.75 ± 1.64**	**76.00 ± 1.41** → 97.00 ± 2.83	
antmaze-medium-diverse-v2	62.00 ± 4.30 → 98.25 ± 1.48	59.40 ± 17.83 → **98.6 ± 1.67**	
antmaze-large-play-v2	31.75 ± 8.87 → 97.25 ± 1.79	**71.6 ± 5.32** → **97.8 ± 1.30**	
antmaze-large-diverse-v2	44.00 ± 8.69 → 91.50 ± 1.79	**73.4 ± 6.35** → **97.8 ± 1.48**	
Average	53.04 → 97.33	**78.69** → **98.19**	

a) The best scores in offline pre-training and online fine-tuning are in bold.

for reward gaps and unsuitable data sampling methods.

6.3 Performance when rewards do not change

We also conduct experiments to compare ESR-O2O with other offline-to-online methods that focus on the fine-tuning process within a single task. For this purpose, we choose the challenging navigation task Antmaze from D4RL, which is a widely-used benchmark that does not contain reward changes. The difficulty of this task lies in the sparse rewards and the need of exploration. Baselines including CQL, IQL, AWAC, and Cal-QL [10] are adopted. The results of the comparisons are presented in Table 2. Among the baselines, IQL shows the best offline performance, while Cal-QL achieves the best fine-tuning performance on most tasks. However, ESR-O2O outperforms all these methods by a significant margin, especially in the case of 'antmaze-large' tasks with the most complex environments.

6.4 Ablation study

To demonstrate the feasibility of our proposed design, we examine the impact of ensembles applied on SR and Q networks on the performance. First, we compare our method with two variants by eliminating ensembles of SR (indicated by 'W/o ensemble SR') and all ensembles (indicated by 'W/o ensemble SR & Q'). In the variant without ensemble SR and Q, ESR-O2O essentially becomes the original version of the successor representation as described in Section 4. This variant only incorporates layer normalization to mitigate overestimation during offline pre-training. Second, we evaluate the effect of the number of ensemble networks when n ranges from 2 to 10. We present the ablation results in Figures 9 and 10. These results illustrate that the ensemble of SR is critical for fine-tuning, especially for large reward gaps. The ensemble of Q networks also plays a significant role when dealing with scenarios with a small reward gap or when the offline data is abundant. As for the quantity of ensemble networks, we observe that our method exhibits strong performance when n is 6 or greater. We refer to Appendix C for more details.

7 Conclusion

In this paper, we have proposed a novel method incorporating ensemble networks and successor representations to handle the task generalization problem in the offline-to-online RL setting. The integration of ensemble networks allows our model to capture diverse representations and reward functions of the environment. Through extensive experiments on various benchmark tasks, we have demonstrated significant improvements in the agent's ability to transfer knowledge from offline data to online environments. Our method outperforms state-of-the-art techniques and provides superior generalization performance. Our work contributes to the broader field of RL by addressing a fundamental limitation of offline-to-online learning. The proposed method opens up new avenues for real-world applications where collecting online data is expensive or time-consuming, with the potential to handle task generalization.

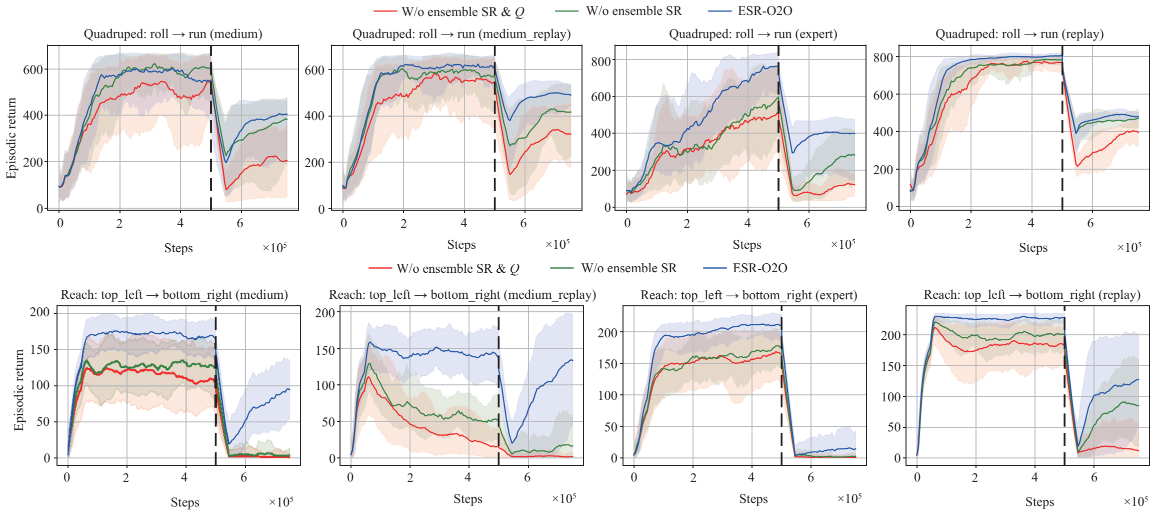

Figure 9 (Color online) Ablation on ensemble SR and ensemble Q. The shaded area indicates the variance across 5 random seeds.

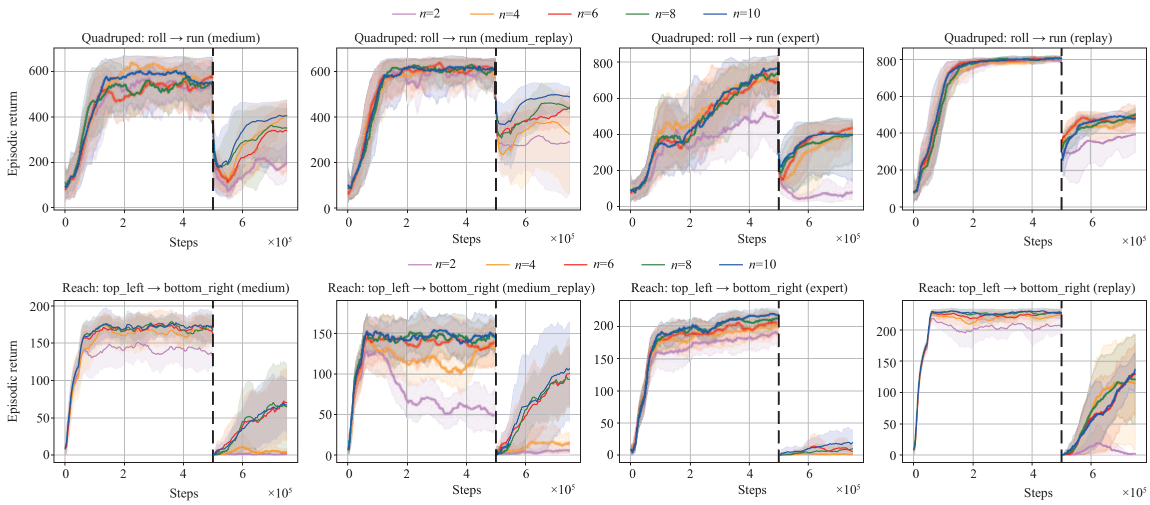

Figure 10 (Color online) Ablation on the quantity of ensemble networks n. The shaded area indicates the variance across 3 random seeds.

In the fine-tuning phase, we have chosen to fix the representation networks to prevent deviation caused by distributional shifts. However, it is worth exploring the possibility of fine-tuning the representation networks using new experiences. Techniques like model expansion [11] could also be employed to further stabilize the representations in the face of distributional shifts. Additionally, investigating more effective measurements for reward gaps is important. Kullback-Leibler or Jensen-Shannon divergences can be used to characterize the gap between reward distributions, and diffusion models can serve as an efficient distribution estimator [56].

Acknowledgements This work was supported by National Science Fund for Distinguished Young Scholars (Grant No. 62025602), National Natural Science Foundation of China (Grant Nos. 62306242, U22B2036, 11931015), Fok Ying-Tong Education Foundation China (Grant No. 171105), Tencent Foundation, XPLORER PRIZE, Science Center Program of National Natural Science Foundation of China (Grant No. 62188101), and Heilongjiang Touyan Innovation Team Program.

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge. Nature, 2017, 550: 354–359
2 Lu X Z, Jie J F, Lin Z H, et al. Reinforcement learning based energy efficient robot relay for unmanned aerial vehicles against smart jamming. Sci China Inf Sci, 2022, 65: 112304

3 Liu N J, Lu T, Cai Y H, et al. Manipulation skill learning on multi-step complex task based on explicit and implicit curriculum learning. Sci China Inf Sci, 2022, 65: 114201
4 Chen S T, Jian Z Q, Huang Y H, et al. Autonomous driving: cognitive construction and situation understanding. Sci China Inf Sci, 2019, 62: 081101
5 Chen H, Yuan K, Huang Y J, et al. Feedback is all you need: from ChatGPT to autonomous driving. Sci China Inf Sci, 2023, 66: 166201
6 Levine S, Kumar A, Tucker G, et al. Offline reinforcement learning: tutorial, review, and perspectives on open problems. 2020. ArXiv:2005.01643
7 Nair A, Gupta A, Dalal M, et al. AWAC: accelerating online reinforcement learning with offline datasets. 2020. ArXiv:2006.09359
8 Lee S, Seo Y, Lee K, Abbeel P, et al. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In: Proceedings of Conference on Robot Learning, Auckland, 2022. 1702–1712
9 Kostrikov I, Nair A, Levine S. Offline reinforcement learning with implicit Q-learning. In: Proceedings of the International Conference on Learning Representations, 2022
10 Nakamoto M, Zhai Y, Singh A, et al. Cal-QL: calibrated offline RL pre-training for efficient online fine-tuning. 2023. ArXiv:2303.05479
11 Zhang H, Xu W, Yu H. Policy expansion for bridging offline-to-online reinforcement learning. 2023. ArXiv:2302.00935
12 Kulkarni T, Saeedi A, Gautam S, et al. Deep successor reinforcement learning. 2016. ArXiv:1606.02396
13 Barreto A, Dabney W, Munos R, et al. Successor features for transfer in reinforcement learning. In: Proceedings of the Advances in Neural Information Processing Systems, 2017. 30
14 Wang Z, Mu C, Hu S, et al. Modelling the dynamics of regret minimization in large agent populations: a master equation approach. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, Vienna, 2022. 23–29
15 Chu C, Li Y, Liu J, et al. A formal model for multiagent q-learning dynamics on regular graphs. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2022. 194–200
16 Li X X, Peng Z H, Jiao L, et al. Online adaptive Q-learning method for fully cooperative linear quadratic dynamic games. Sci China Inf Sci, 2019, 62: 222201
17 Fujimoto S, Meger D, Precup D. Off-policy deep reinforcement learning without exploration. In: Proceedings of the International Conference on Machine Learning, 2017. 2052–2062
18 Wu Y, Tucker G, Nachum O. Behavior regularized offline reinforcement learning. 2019. ArXiv:1911.11361
19 Fujimoto S, Gu S. A minimalist approach to offline reinforcement learning. In: Proceedings of the Advances in Neural Information Processing Systems, 2021. 20132–20145
20 Kumar A, Zhou A, Tucker G, et al. Conservative Q-learning for offline reinforcement learning. In: Proceedings of the Advances in Neural Information Processing Systems, 2020. 1179–1191
21 An G, Moon S, Kim J, et al. Uncertainty-based offline reinforcement learning with diversified Q-ensemble. In: Proceedings of the Advances in Neural Information Processing Systems, 2021
22 Ghasemipour S, Gu S, Nachum O. Why so pessimistic? Estimating uncertainties for offline RL through ensembles, and why their independence matters. In: Proceedings of the Advances in Neural Information Processing Systems, Louisiana, 2022
23 Ball P, Smith L, Kostrikov I, et al. Efficient online reinforcement learning with offline data. 2023. ArXiv:2302.02948
24 Beeson A, Montana G. Balancing policy constraint and ensemble size in uncertainty-based offline reinforcement learning. 2023. ArXiv:2303.14716
25 Dayan P. Improving generalization for temporal difference learning: the successor representation. Neural Computation, 1993, 5: 613–624
26 Uchendu I, Xiao T, Lu Y, et al. Jump-start reinforcement learning. 2022. ArXiv:2204.02372
27 Zhao Y, Boney R, Ilin A, et al. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. 2022. ArXiv:2210.13846
28 Zhao K, Ma Y, Liu J, et al. Improving offline-to-online reinforcement learning with Q-ensembles. In: Proceedings of ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems, 2023
29 Li J, Hu X, Xu H, et al. PROTO: iterative policy regularized offline-to-online reinforcement learning. 2023. ArXiv:2305.15669
30 Ball P, Lu C, Parker-Holder J, et al. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment. In: Proceedings of the International Conference on Machine Learning, 2021. 619–629
31 Xu K, Bai C, Ma X, Wang D, et al. Cross-domain policy adaptation via value-guided data filtering. 2023. ArXiv:2305.17625
32 Mazoure B, Kostrikov I, Nachum O, et al. Improving zero-shot generalization in offline reinforcement learning using generalized similarity functions. In: Proceedings of the Advances in Neural Information Processing Systems, 2022. 25088–25101
33 Ying C, Hao Z, Zhou X, et al. Reward informed dreamer for task generalization in reinforcement learning. 2023. ArXiv:2303.05092
34 Zhang J, Springenberg J, Boedecker J, et al. Deep reinforcement learning with successor features for navigation across similar environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots And Systems (IROS), Vancouver, 2017. 2371–2378
35 Abdolshah M, Le H, George T, et al. A new representation of successor features for transfer across dissimilar environments. In: Proceedings of the International Conference on Machine Learning, 2021. 1–9
36 Momennejad I, Russek E M, Cheong J H, et al. The successor representation in human reinforcement learning. Nat Hum Behav, 2017, 1: 680–692
37 Touati A, Ollivier Y. Learning one representation to optimize all rewards. In: Proceedings of the Advances in Neural Information Processing Systems, 2021. 13–23
38 Touati A, Rapin J, Ollivier Y. Does zero-shot reinforcement learning exist? In: Proceedings of the 11th International Conference on Learning Representations, Kigali, 2023
39 Osband I, Blundell C, Pritzel A, et al. Deep exploration via bootstrapped DQN. In: Proceedings of the Advances in Neural Information Processing Systems, Barcelona, 2016
40 Chen R, Sidor S, Abbeel P, et al. UCB exploration via Q-ensembles. 2017. ArXiv:1706.01502
41 Chen X, Wang C, Zhou Z, et al. Randomized ensembled double Q-learning: learning fast without a model. In: Proceedings of the International Conference on Learning Representations, 2021
42 Bai C, Wang L, Han L, et al. Principled exploration via optimistic bootstrapping and backward induction. In: Proceedings of International Conference on Machine Learning, 2021. 577–587
43 Qiu S, Wang L, Bai C, et al. Contrastive UCB: provably efficient contrastive self-supervised learning in online reinforcement

learning. In: Proceedings of International Conference on Machine Learning, 2022. 18168–18210
44 Bai C, Wang L, Han L, et al. Dynamic bottleneck for robust self-supervised exploration. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 17007–17020
45 Wen X, Yu X, Yang R, et al. Towards robust offline-to-online reinforcement learning via uncertainty and smoothness. 2023. ArXiv:2309.16973
46 Fort S, Hu H, Lakshminarayanan B. Deep ensembles: a loss landscape perspective. 2019. ArXiv:1912.02757
47 Lan Q, Pan Y, Fyshe A, et al. Maxmin Q-learning: controlling the estimation bias of Q-learning. In: Proceedings of the International Conference on Learning Representations, Addis Ababa, 2020
48 Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: Proceedings of the International Conference on Machine Learning, Stockholm, 2018. 1587–1596
49 Blier L, Tallec C, Ollivier Y. Learning successor states and goal-dependent values: a mathematical viewpoint. 2021. ArXiv:2101.07123
50 Abbasi-Yadkori Y, Pál D, Szepesvári C. Improved algorithms for linear stochastic bandits. In: Proceedings of the Advances in Neural Information Processing Systems, 2011
51 Jin Y, Yang Z, Wang Z. Is pessimism provably efficient for offline RL? In: Proceedings of the International Conference on Machine Learning, 2021. 5084–5096
52 Bai C, Wang L, Yang Z, et al. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In: Proceedings of the International Conference on Learning Representations, 2022
53 Bai C, Wang L, Hao J, et al. Pessimistic value iteration for multi-task data sharing in offline reinforcement learning. Artif Intell, 2024, 326: 104048
54 Yu T, Quillen D, He Z, et al. Meta-world: a benchmark and evaluation for multi-task and meta reinforcement learning. In: Proceedings of Conference on Robot Learning, 2020. 1094–1100
55 Fu J, Kumar A, Nachum O, et al. D4RL: datasets for deep data-driven reinforcement learning. 2020. ArXiv:2004.07219
56 Oko K, Akiyama S, Suzuki T. Diffusion models are minimax optimal distribution estimators. In: Proceedings of the International Conference on Machine Learning, 2023. 26517–26582