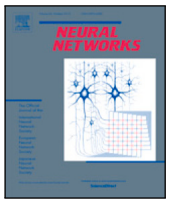
Neural Networks 187 (2025) 107342

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Rethinking exploration–exploitation trade-off in reinforcement learning via cognitive consistency

Da Wang [a], Wei Wei [a,*], Lin Li [a], Xin Wang [b], Jiye Liang [a]

[a] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education and the School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China
[b] Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Off-policy reinforcement learning
Sample efficiency
Exploration–exploitation trade-off
Cognitive consistency

ABSTRACT

The exploration–exploitation dilemma is one of the fundamental challenges in deep reinforcement learning (RL). Agents must strike a trade-off between making decisions based on current beliefs or gathering more information. Prior work mostly prefers devising sophisticated exploration methods to ensure accurate target Q-values or learn rewards and actions association, which may not be intelligent enough for sample efficiency. In this paper, we propose to rethink the trade-off between exploration and exploitation from the perspective of cognitive consistency: humans tend to think and behave in line with their existing knowledge structures (maintaining cognitive consistency), yielding satisfactory results within a brief timeframe. We argue that maintaining consistency, specifically through pessimistic exploration, within the context of optimal policy-oriented cognition, can improve efficiency without compromising performance. To this end, we propose a Cognitive Consistency (CoCo) framework. CoCo first leverages a self-imitating distribution correction approach to pursue cognition oriented toward the optimal policy. Then, it conservatively implements pessimistic exploration by extracting novel inconsistency-minimization objectives inspired by label distribution learning. We validate our framework across various standard off-policy RL tasks and show that maintaining cognitive consistency improves sample efficiency and performance. Code is available at https://github.com/DkING-lv6/CoCo.

1. Introduction

Reinforcement learning (RL) (Sutton & Barto, 2018) has shown great promise in various domains, such as games (Mnih et al., 2015; Ye, Liu, Kurutach, Abbeel, & Gao, 2021), robotics (Leottau, del Solar, & Babuška, 2018; Levine, Finn, Darrell, & Abbeel, 2016), and realistic simulated environments (Hsu, Ren, Nguyen, Majumdar, & Fisac, 2023; Schulman, Levine, Abbeel, Jordan, & Moritz, 2015). The trade-off between exploration and exploitation is a fundamental problem in RL and online decision-making. Agents need to strike a balance between trying new behaviors (exploration) for gathering more information and making decisions utilizing current beliefs (exploitation). Previous work (Ecoffet, Huizinga, Lehman, Stanley, & Clune, 2021; Han & Sung, 2021; Mavor-Parker, Young, Barry, & Griffin, 2022; Yuan, Pun, & Wang, 2022; Zhang et al., 2021) concentrates on providing intrinsic rewards or balancing actions to scale up to larger state spaces, achieving great success in sparse-reward and hard-exploration environments. Recent studies (Liu et al., 2023; Sun et al., 2022; Yang et al., 2023) commonly adopt optimistic exploration and pessimistic exploitation to address the dilemma.

However, frequently neglected is the reality that optimistic exploration, despite its widely used, is not the most effective strategy and is indeed suboptimal in terms of efficiency. Most heuristic methods teach agents often need to learn strategies "diligently" or even "inch by inch" to achieve the desired performance, which may not be "intelligent" enough at sample efficiency. In this paper, we initiate a critical rethinking of the balance between exploration and exploitation. Our emphasis on exploitation is not related to that the high (or low) rewards could lead to profit (or failure). Instead, it is grounded in the premise that when the expected reward improves, the likelihood of it being the optimal action increases, thereby requiring a more refined calculation of its value. In essence, expending computational effort to quantify the extent of a poor policy's inadequacies is unnecessary; however, with policies that show promise, such investment is essential to evaluate their strengths fully, allowing us to single out the most

* Corresponding author.
E-mail addresses: sxu_wd@163.com (D. Wang), weiwei@sxu.edu.cn (W. Wei), lilynn1116@sxu.edu.cn (L. Li), xin_wang@tsinghua.edu.cn (X. Wang), ljy@sxu.edu.cn (J. Liang).

https://doi.org/10.1016/j.neunet.2025.107342
Received 3 November 2024; Received in revised form 10 February 2025; Accepted 1 March 2025
Available online 11 March 2025

effective one. The significance we assign to exploitation is not focused on the accumulation of rewards but on the accurate estimation of the optimal actions.

Based on the above understanding, we propose to employ cognitive consistency when balancing exploration and exploitation: social psychology and real-world experiences show that humans tend to think and behave in line with their existing knowledge structures, that is, to maintain cognitive consistency (Festinger, 1962). This tendency allows them to achieve satisfactory results in a short time. In the context of RL, the core of cognitive consistency lies in conducting pessimistic exploration and optimistic exploitation under reasonable premises. In other words, the priority is to guide the agent in learning an effective policy and then conduct explorations in its vicinity. It is unnecessary to systematically examine poor policies to obtain accurate estimations and confirm their deficiencies (as exemplified through a didactic example in Section 3.1).

Supported by the preceding analysis, the application of cognitive consistency in RL has gained a clearer significance: by integrating cognitive consistency into the trade-off between exploration and exploitation, one can enhance efficiency without compromising performance. To this end, we introduce a Cognitive Consistency (CoCo) framework. Specifically, we first develop a self-imitating distribution correction approach to capture high-yield samples, to pursue cognition oriented toward the optimal policy. Then, we extract a novel inconsistency-minimization objective inspired by label distribution learning (LDL) (Geng, 2016; Wang, Geng, & Xue, 2021) to conservatively implement pessimistic exploration. Finally, we incorporate the above two steps into the coherent CoCo framework through a briefly reweighted, uniformly sampled loss function. As example, we implement cognitive consistency under the actor-critic method for practical use. Extensive experiments show that properly maintaining cognitive consistency can substantially improve sample efficiency and performance of off-policy RL. We also delve into the proposed algorithm and design several experiments, such as ablation studies and tasks with reward noise, to demonstrate some key properties of CoCo.

In summary, our contribution is three-fold:

- **Novel perspective:** We propose to rethink the trade-off between exploration and exploitation from a novel perspective of cognitive consistency. Conducting pessimistic exploration and optimistic exploitation under reasonable premises to improve sample efficiency is our key contribution compared to previous studies.
- **Novel methodology:** We present a novel framework CoCo to address the exploration–exploitation dilemma. Technically, we use self-imitating distribution correction approach to pursue cognition-oriented optimistic exploitation and innovatively introduce an inconsistency minimization objective inspired by LDL to achieve pessimistic exploration. We incorporate the above two steps through a briefly reweighted, uniformly sampled loss function, thereby rendering the implementation of CoCo both straightforward and accessible.
- **Superior performance and insightful results:** We provide didactic studies to support our rethinking and conduct extensive experiments to demonstrate the effectiveness of the proposed CoCo framework. Additionally, we design ablation studies on the key components of CoCo to gain deeper insights into their contributions.

2. Background

This section will briefly introduce the related work and preliminaries. It should be stated that our total algorithm is predicated on a more comprehensive design, wherein exploration is focused on the perspective of sample collection, and exploitation is concerned with the study of how to more effectively utilize the collected samples to train the policy.

2.1. Related work

2.1.1. Exploration

Exploration has long been a critical issue in RL. Prior work (Ecoffet et al., 2021; Han & Sung, 2021; Mavor-Parker et al., 2022; Yuan et al., 2022; Zhang et al., 2021) mostly designs sophisticated exploration technical to provide intrinsic rewards or balancing actions to scale up to larger state spaces. Further study (Liu et al., 2023; Sun et al., 2022; Yang et al., 2023) attempts to introduce pessimistic exploitation into optimistic exploration for gaining better performance. In contrast, we emphasize that modifying the exploration policy (i.e., behavior) directly impacts the experience collection process and ultimately determines the training distribution (Kumara, Gupta, & Levine, 2020). Based on this insight, we propose consolidating the behavior decisions of cognition oriented toward the optimal policy to increase the number of high-yield samples in the replay buffer. It is important to note that we are not discussing our work in the context of improving exploration directly.

2.1.2. Exploitation

The exploitation in off-policy RL is expressed as exploiting past experiences (Oh, Guo, Singh, & Lee, 2018). Experience replay (ER) (Lin, 1992) is a widespread technique in off-policy RL, storing experiences in a replay buffer for reuse. The problem of data utilization in the replay buffer has been widely studied. Prioritization or reweighting of replay samples has achieved great performance in ER methods, with criteria such as TD error (Schaul, Quan, Antonoglou, & Silver, 2016), corrective feedback (Kumara et al., 2020; Lee, Laskin, Srinivas, & Abbeel, 2021) and on-policiness (Liu et al., 2021; Novati & Koumoutsakos, 2019; Sinha, Song, Garg, & Ermon, 2022; Sun, Zhou, & Li, 2020; Wang, Wu, Vuong, & Ross, 2019; Wei, Wang, Li, & Liang, 2024). It has been demonstrated that increasing on-policiness (fixing the data distribution gap between the behavior policy and the current policy) can lead to significant performance improvement. However, this type of method is hampered by the behavior policy. It may be inefficient when the agent pays excessive attention to the low-yield region exploration. In this work, we analyze the pathological concerns associated with the on-policiness priority criterion and correct it using the distribution of cognition oriented toward the optimal policy. More detailed related works are listed in Appendix A.

2.2. Preliminaries

A reinforcement learning problem can be described as training a policy in an infinite-horizon, discounted Markov decision process (MDP) denoted as $(\mathcal{S}, \mathcal{A}, P, r, \gamma, p_0)$, where $\mathcal{S}$ and $\mathcal{A}$ represent state and action spaces. $P(s'|s,a)$ and $r(s,a)$ are the transition and reward function. $\gamma \in (0,1)$ is the discount factor and $p_0(s)$ is the distribution of the initial state. The goal is to find an optimal policy that maximizes the expected cumulative discounted reward $J(\pi) := \mathbb{E}_\pi[\Sigma_{t=0}^{\infty}\gamma^t r(s_t,a_t)]$, where the expectation is over trajectories sampled from $s_0 \sim \rho_0, a_t \sim \pi(\cdot \mid s_t)$, and $s_{t+1} \sim P(\cdot \mid s_t, a_t)$ for $t \geq 0$. We denoted the discounted stationary state distribution of the policy $\pi(a \mid s)$ as $d^\pi(s)$ and the corresponding state–action distribution as $d^\pi(s,a) = d^\pi(s)\pi(a \mid s)$. Then, we can rewrite $J(\pi) = \mathbb{E}_{d^\pi}[r(s,a)]$.

For any stationary policy π, a standard definition of the state value function is defined as $V^\pi(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)]$, and its corresponding state–action value function, or Q-function as $Q^\pi(s,a) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t) \mid s_0 = s, a_0 = a\right]$. The goal is to learn an optimal approximation to the Q-function (i.e. $Q^*(s,a)$) by applying successive Bellman projections. $Q^*(s,a)$ satisfies the Bellman equation $Q^*(s,a) = \mathcal{B}^*(Q^*(s,a))$, where $\mathcal{B}^*$ denote the Bellman optimal operator $(\mathcal{B}^*Q)(s,a) = r(s,a) + \gamma E_{s'\sim P}\left[\max_{a'} Q(s',a')\right]$.

It turns out that considering the replay of the prioritized experience as the selection of a favorable prioritization distribution is plausible (Sinha et al., 2022). Given a replay buffer D and the corresponding

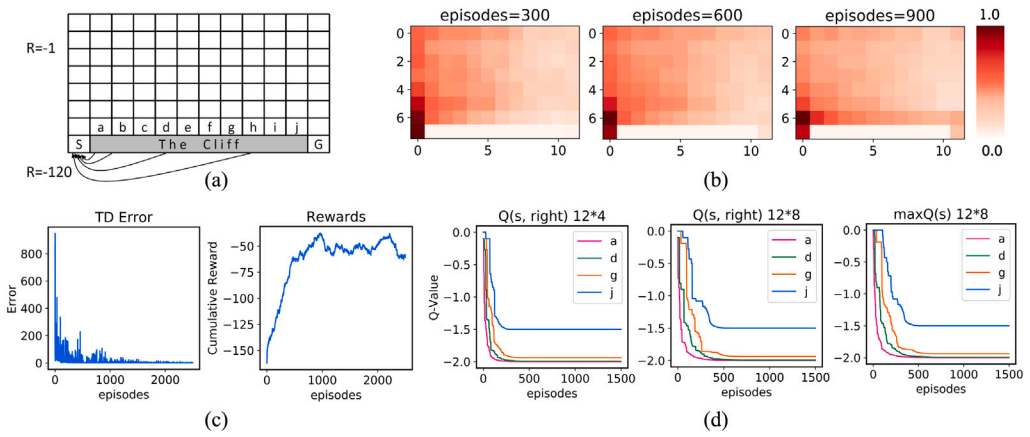


Fig. 1. A Cliff sample. (a) S and G are start and goal states, the actions are *top*, *down*, *left* and *right*. Reward is -1 on each step, stepping into the region "Cliff" incurs a reward of -120 and sends the agent instantly back to the start (the current episode does not end, only when the agent walks enough 100 steps or reaches the end state). (b) The visualized frequency of state visits at different episodes. The number of visits is normalized and is visualized by the color of the grid. (c) TD error (absolute value) and cumulative reward (smoothed) at each training episode. (d) Variation of Q-values at key states a, d, g and j.

data distribution d^{μ}, one could train the Q-network with parameters θ by optimizing the following loss:

$$L_Q(\theta; d^{\mu}, \omega) = \mathbb{E}_{d^{\mu}}[\omega(s,a)(Q_\theta(s,a) - \mathcal{B}^* Q_\theta(s,a))^2], \tag{1}$$

where $\omega : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^+$ is the prioritization weights of replayed samples and $\mathcal{B}^*Q$ refers to the *target value* for the projection step.

Soft Actor-Critic (SAC) (Haarnoja, Zhou, Abbeel, & Levine, 2018) is a representative off-policy actor-critic RL algorithm. It introduces the entropy into the optimization objective:

$$J(\pi) = \mathbb{E}_{s\sim P, a\sim\pi} \sum_{t=0}^{T} \left[r\left(s_t, a_t\right) + \alpha \mathcal{H}\left(\pi\left(\cdot \mid s_t\right)\right)\right], \tag{2}$$

where α is the temperature coefficient, $\mathcal{H}(\pi(\cdot \mid s))$ is the entropy of policy π at state s. By optimizing the objective function, the agent is encouraged to act as stochastically as possible while maximizing the cumulative gain, thus improving the robustness and stability of the policy.

3. Our method

In this section, we introduce the CoCo framework. We first illustrate a potential issue that affects the sample efficiency of optimistic exploration, using a didactic example (Section 3.1). Secondly, we propose a self-imitating distribution correction approach to pursue cognition-oriented optimistic exploitation (Section 3.2). Thirdly, we analyze the inconsistency issue in RL from the perspective of LDL, then introduce an inconsistency minimization objective to achieve pessimistic exploration (Section 3.3). Finally, we integrate the above two components into the coherent CoCo framework and show the mutual benefit between them (Section 3.4).

3.1. A didactic example for the motivation

The requirements of RL for an optimal policy vary depending on the type of task. As for the multi-goal task, the end goal of each episode is randomly generated. It is always necessary to learn the policies of critical state regions, even "carpet" training, to consistently get good performance. In contrast, for some tasks with fixed goal positions, the optimal policy may prefer to be unique, i.e., there are low-yield regions in the state space. Intuitively, the learned approximate value function training on the distribution induced by the current policy π will not help much in consolidating the optimal policy π^* when π is far from the π^*. While the behavior policy is influenced by this value function, it might exacerbate the generation of these useless samples in the next iteration, leading to a pathological concern on the low-yield regions. This phenomenon is severe in the over-exploitation caused by optimistic exploration, especially using the on-policiness criterion. Note that the samples on these non-optimal paths are not completely useless, and they may allow some edge states to be estimated accurately. However, the accurate estimates may not affect the ranking of the corresponding Q-value, which may be more important for optimal decision-making.

For the above case, we design a modified classical Cliff example to verify our intuition, as shown in Fig. 1. To reflect the redundancy of the state space, we expand the "safe region" of the cliff environment by increasing the original $12 * 4$ (Sutton & Barto, 2018) range to $12 * 8$ in Fig. 1(a). In the experiments, experience replay is performed with the on-policiness prioritization criterion, and the value function is updated using Q-learning (Sutton & Barto, 2018).

In the Cliff task, the optimal policy is unique, starting at S and ending at G, with pathway states $a - j$. From the heat map of state visits frequency in Fig. 1(b), it is clear that most of the exploration interactions occur in the upper left region before starting to focus on the optimal path. Fig. 1(c) shows the corresponding TD error and cumulative reward at each training episode. It can be seen that the value function converges at around 2000 episodes, indicating that until then, the agent has been busy updating the Q-value of the explored regions under the influence of over-exploitation (on-policiness). The overall error is unstable frequently, although it shows a decreasing trend. This verifies the effect of Bellman error accumulation on value function update, i.e., the Q-value of samples replayed with high frequency (close to the start S) is difficult to get convergence rapidly.

Our intuition is verified in Fig. 1(d). We observe the Q-value at selected key states a, d, g, and j to investigate the learning of the optimal policy. Comparing the first two columns, the convergence value $Q(s, right)$ of the optimal decision *right* for each state in $12 * 8$ is equal to the counterpart in $12 * 4$. It indicates that the low-yield region fails to affect the Q-value of the optimal decision. In addition, the comparison of the last two columns reflects the difference between $Q(s, right)$ and the optimal decision $maxQ(s)$. It shows that the $Q(s, right)$ converges at about 400 episodes and keeps in line with the current optimal decision. When we combine Fig. 1(b) with our observations, it becomes evident that the optimal path is only visited a few times. Nonetheless, its value can be updated with precision. Most of the exploration and updates are inefficiently spent on low-yield regions, as we suspected. Focusing on low-yield regions only updates their value estimate, but it does not enhance the learning of the optimal policy. Therefore, we should pay less attention to low-yield regions, especially those that will almost never be visited again eventually.

The above example shows that paying too much emphasis on low-yield regions can diminish sample efficiency. The over-exploitation

caused by optimistic exploration, especially using the on-policiness criterion hampered by behavior policies, can further exacerbate the problem.

3.2. Self-imitating distribution correction

The case studies have illuminated that exploitation should be centered around the optimal policy π^*. Establishing this as a reasonable premise is crucial for successfully applying cognitive consistency in RL. Since the optimal policy π^* cannot be accessed in advance, we develop a tractable approximation in the following claim.

Claim 3.1. *Within the entropy-regularized off-policy RL framework, the policies with higher Monte-Carlo* ***episode returns*** *are closer to the optimal policy.*

The cognition refers to the knowledge (i.e., the policy) that an agent utilize. To support the above claim, we introduce the relationship between the state–action distribution and the gap in policies' cumulative rewards (Lemma 3.2). Then, we use the lower bound of the optimal soft Q-value (Lemma 3.3) to prove our claim.

Lemma 3.2. *Assume that reward function is bounded in absolute value $R_{\max}$. For any two policies $\tilde{\pi}$ and π, the gap in policies' cumulative rewards is bounded by the state–action distribution discrepancy,*

$$|J(\tilde{\pi}) - J(\pi)| \leq \frac{2R_{\max}}{1-\gamma} D_{\mathrm{TV}}\left(d^{\tilde{\pi}}(s,a), d^{\pi}(s,a)\right), \tag{3}$$

where $D_{\mathrm{TV}}(P,Q) = \frac{1}{2}\|P-Q\|_1$ is the total variation (TV) distance of distribution P and Q.

Lemma 3.2 is part of the *error-propagation framework* proposed in Xu, Li, and Yu (2021). As the goal of RL is to find a policy that maximizes $J(\pi) := \mathbb{E}_{\pi}[\Sigma_{t=0}^{\infty}\gamma^t r(s_t,a_t)]$, the optimization problem can be expressed as minimize $J(\pi^*) - J(\mu)$ (Liu et al., 2021), where π^* is the optimal policy. Considering that the update of the policy is influenced by the actual replayed samples (i.e., corresponding to the behavioral policy μ). The behavior policy μ refers to the strategy that an agent actually uses when interacting with the environment ($\mu \neq \pi$ in off-policy RL). From Lemma 3.2, we have that,

$$\left|J(\pi^*) - J(\mu)\right| \leq \frac{2R_{\max}}{1-\gamma} D_{\mathrm{TV}}\left(d^{\pi^*}(s,a), d^{\mu}(s,a)\right). \tag{4}$$

Note that Xu et al. (2021) studies error propagation in behavioral cloning, while we emphasize the correlation between performance and distribution. Both tell us that the state–action discrepancy plays an essential role in analyzing the gap in policies' cumulative rewards. Eq. (4) provides theoretical guidance for the prioritization of replay distribution and is the key to establishing a connection between the goal of RL and the requirement for replayed samples. Furthermore, $D_{\mathrm{TV}}\left(d^{\pi^*}(s,a), d^{\mu}(s,a)\right)$ represents a mismatch between the data distribution d^{μ} and that of the optimal policy π^*. That is, the samples similar to optimal distribution should be paid more attention to, i.e., weighting by $\omega(s,a) := d^{\pi^*}(s,a)/d^{\mu}(s,a)$. This result is consistent with our proposition:

$$\omega(s,a) := \frac{d^{\pi^*}(s,a)}{d^{\mu}(s,a)} = \underbrace{\frac{d^{\pi}(s,a)}{d^{\mu}(s,a)}}_{(a)} \cdot \underbrace{\frac{d^{\pi^*}(s,a)}{d^{\pi}(s,a)}}_{(b)}, \tag{5}$$

where term (a) is the on-policiness weight and term (b) is the distribution gap between the current policy π and the optimal policy π^*. The term (b) is considered in this paper as a correction to alleviate the pathological concerns discussed in Section 3.1.

The aforementioned analysis illustrates the correlation between the replayed distribution and the gap in policies' cumulative rewards, yet it does not capture the monotonicity that is reflected in Claim 3.1. Our next step is to supplement the proof of Claim 3.1 with the ultimate goal of RL and the lower bound of the optimal soft Q-value.

Lemma 3.3 (*Lower Bound of Optimal Soft Q-Value*)**.** *Let π^* be an optimal policy in entropy-regularized RL: $\pi^* = \mathrm{argmax}_{\pi}\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^t\left(r_t + \alpha\mathcal{H}_t^{\pi}\right)\right]$, where $\mathcal{H}_t^{\pi} = -\log\pi\left(a_t \mid s_t\right)$ is the entropy of the policy π, and $\alpha \geq 0$ represents the weight of entropy bonus. It is straightforward that the expected return of any behavior policy μ can serve as a lower bound of the optimal soft Q-value as follows:*

$$\begin{aligned} R_{\tau}^{\pi^*} &= \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t\left(r_t + \alpha\mathcal{H}_t^{\pi^*}\right)\right] \\ &\geq \mathbb{E}_{\mu}\left[\sum_{t=0}^{\infty}\gamma^t\left(r_t + \alpha\mathcal{H}_t^{\mu}\right)\right] \end{aligned} \tag{6}$$

because the entropy-regularized return of the optimal policy is always greater or equal to that of any other policies.

Since the optimal policy π^* cannot be accessed in advance, we develop a tractable approximation for $d^{\pi^*}(s,a)/d^{\mu}(s,a)$. Inspired by Self-Imitation Learning (SIL) (Oh et al., 2018) – which learns to imitate state–action pairs in the replay buffer only when the return $R_t = \sum_{k=t}^{\infty}\gamma^{k-t}r_k$ in the past is greater than the agent's value estimate. Differently, we consider the ultimate goal of RL (Liu et al., 2021) and evaluate the sample $(s,a) \sim \tau$ with the total return $R_{\tau}^{\pi} = \Sigma_{t=0}^{T}\gamma^t r(s_t,a_t)$ (i.e., the Monte-Carlo **episode return**) of trajectory $\tau = \{s_t,a_t\}_{t=0}^{T}$. We then define the self-imitating policy π^{si} and obtain the relationship by Lemma 3.3:

$$R_{\tau}^{\pi^*} \geq R^{\mathrm{MAX}} = R_{\tau}^{\pi^{\mathrm{si}}} \geq R_{\tau}^{\pi}, \tag{7}$$

where R^{MAX} denotes the highest return the agent has received so far, i.e. the episode return of π^{si}. Eq. (7) indicates the difference between $J(\pi^*)$ and $J(\pi)$ is positively correlated with the episode return under the corresponding data distribution.

By combining Eq. (7) with Eq. (4), we can conclude that in a common entropy-regularized off-policy RL framework, the state action pair with a larger Monte-Carlo episode return is closer to π^* and thus more authoritative. So far, Claim 3.1 has been proven. Furthermore, we can also deduce that a relative upper bound during the training is R^{MAX}.

3.2.1. A practical implementation

Based on the above observations, to pursue the cognition oriented toward the optimal policy, it is necessary to keep the replay distribution used for training close to that of π^{si}. To this end, we consider a self-imitating distribution correction approach to weight samples, which focuses on those generated by the historical optimal policy π^{si}. Once R_{τ}^{π} exceeds $R_{\tau}^{\pi^{\mathrm{si}}}$, it is immediately replaced. Thus, π^{si} can always play the role of a guide whose quality of guidance depends on the total variation distance $D_{TV}(\pi^{\mathrm{si}},\pi^*)$. Now, we can give the self-imitating weight:

$$\omega^{\mathrm{si}}(s,a) := \frac{d^{\pi^{\mathrm{si}}}(s,a)}{d^{\mu}(s,a)}. \tag{8}$$

To calculate $d^{\pi^{\mathrm{si}}}(s,a)/d^{\mu}(s,a)$, which is essentially an importance weight, we draw on the Likelihood-Free Importance Weighting (LFIW) (Sinha et al., 2022). First, we add a small size buffer $\mathcal{D}_{\mathrm{si}}$ to store the samples generated by the historical optimal policy. Second, these samples are used to train a weight model κ_{ψ} by:

$$L_{\kappa}(\psi) := \mathbb{E}_{\mathcal{D}}\left[f^*\left(f'\left(\kappa_{\psi}(s,a)\right)\right)\right] - \mathbb{E}_{\mathcal{D}_{\mathrm{si}}}\left[f'\left(\kappa_{\psi}(s,a)\right)\right], \tag{9}$$

where f' and f^* is the derivative and convex conjugate of function f. Then, we apply the optimal κ_{ψ} (which outputs $\kappa_{\psi}(s,a)$ for the samples drawn from the conventional buffer $\mathcal{D}$) to estimating the density ratio $\omega^{\mathrm{si}}(s,a)$. The updated κ_{ψ} can be effectively used to estimate the density ratio. This conclusion is supported by the following Lemma (Nguyen, Wainwright, & Jordan, 2010):

Lemma 3.4. *For any convex, lower-semicontinuous function $f : [0,\infty) \to \mathbb{R}$ satisfying $f(1) = 0$, assume existing two probabilistic measures $P, Q \in$*

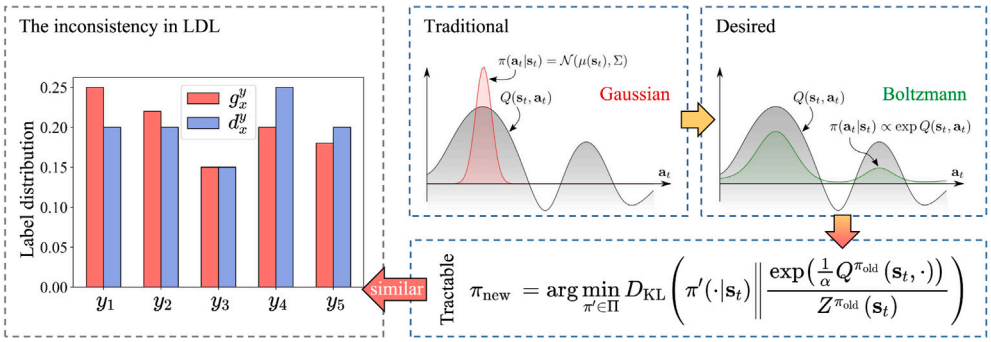


Fig. 2. The inconsistency in RL.

$\mathcal{P}(\mathcal{X})$ *that* $P \ll Q$, *and* $\kappa : \mathcal{X} \to \mathbb{R}^+$,

$$D_f(P \parallel Q) \geq \mathbb{E}_P \left[f'(\kappa(\boldsymbol{x})) \right] - \mathbb{E}_Q \left[f^* \left(f'(\kappa(\boldsymbol{x})) \right) \right], \tag{10}$$

the equality is achieved when $\kappa = \mathrm{d}P/\mathrm{d}Q$.

In Lemma 3.4, $D_f(P \parallel Q) = \int_{\mathcal{X}} f(\mathrm{d}P(\boldsymbol{x})/\mathrm{d}Q(\boldsymbol{x}))\mathrm{d}Q(\boldsymbol{x})$ is the f-divergences (Csiszár, 1964). From the Lemma, we can estimate the density ratio $d^{\pi}(s,a)/\mu(s,a)$. Thus obtain $\omega^{\text{si}}(s,a)$ by minimizing the objective $L_{\kappa}(\psi)$ (Eq. (9)).

The $\omega^{\text{si}}(s,a)$ indicates that we also need to focus on a distance between the recent experience and the optimal policy π^*. It is concluded that self-imitating distribution correction attempts to move closer and closer towards the optimal policy.

3.2.2. The difference between self-imitating distribution correction and prioritizing on-policiness criterion

Comparatively, the former is more comprehensive than the latter. For one thing, self-imitating distribution correction can accurately focus on the sample distribution of the current policy when it works well. Thus, it fully exploits the positive effects of on-policiness. For another, when the current policy performs poorly, on-policiness then falls into a pathological concern. However, our method can shift the focus from on-policy sample distribution to the past good ones. Consequently, it can avoid the long-term performance slump caused by concentrating solely on on-policy experiences.

3.3. Inconsistency minimization for pessimistic exploration

In cognitive dissonance theory (Festinger, 1962), when perceptions are inconsistent, individuals experience a sense of discomfort, prompting them to modify their behavior to alleviate this sensation by rendering their cognitions more consistent. Maintaining cognitive consistency in RL corresponds to minimizing the entropy of the decision distribution in information theory – the higher the entropy, the more hesitant the decision and the more likely it is that cognitive dissonance will occur. Unfortunately, directly pursuing consistency (minimum entropy) is risky since an inappropriate exploration policy can easily cause incorrect value estimation, suboptimal convergence, and other problems. We propose to gradually consolidate cognitive consistency by continuously reducing inconsistency, thereby reducing the impact on exploration. In this subsection, we first analyze the inconsistency issue in RL from the perspective of Label Distribution Learning (LDL) (Geng, 2016; Wang et al., 2021), then introduce an inconsistency minimization objective to consolidate cognitive consistency. To our knowledge, this initiative is unexplored in previous RL literature.

3.3.1. The inconsistency in LDL

The goal of training an LDL model is to learn the whole distribution. While in the test phase, only the top label predicted is needed. That is, LDL may neglect the top label for the sake of learning the whole label distribution, which likely leads to objective inconsistency. Fig. 2 (left) shows the characteristics of label distribution that may result in the inconsistency. The trained distribution has much similar to the true label distribution, but the top label are not consistent, i.e. $\hat{g}(\boldsymbol{x}) \neq \hat{d}(\boldsymbol{x})$ ($\hat{g}(\boldsymbol{x}) = y_1$ and $\hat{d}(\boldsymbol{x}) = y_4$ are the respective top label).

3.3.2. The inconsistency in RL

Essentially, RL can also be considered as a LDL which learns an action distribution under each state. The inconsistency of RL exists mainly in the actor-critic method. Soft Actor-Critic (SAC) (Haarnoja et al., 2018) is a representative off-policy actor-critic RL algorithm. It introduces the entropy into the optimization:

$$J(\pi) = \mathbb{E}_{s \sim P, a \sim \pi} \sum_{t=0}^{T} \left[r\left(s_t, a_t\right) + \alpha \mathcal{H}\left(\pi\left(\cdot \mid s_t\right)\right) \right], \tag{11}$$

where α is the temperature coefficient, $\mathcal{H}(\pi(\cdot \mid s))$ is the entropy of policy π at state s. By optimizing the objective function, the agent is encouraged to act as stochastically as possible while maximizing the cumulative gain, thus improving the robustness and stability of the policy.

For the multimodal Q function, the policy trained by the RL algorithm can only converge to a single choice, while the desired one should fit a Boltzmann distribution (see Fig. 2). For more tractable, SAC restricts the policy to some set of policies Π, which can correspond, for example, to a parameterized family of distributions such as Gaussians:

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathrm{D}_{\mathrm{KL}} \left(\pi'\left(\cdot \mid s_t\right) \parallel \frac{\exp\left(Q^{\pi_{\text{old}}}\left(s_t, \cdot\right)\right)}{Z^{\pi_{\text{old}}}\left(s_t\right)} \right), \tag{12}$$

where $Z^{\pi_{\text{old}}}(s_t)$ normalizes the distribution. We can observe that $\pi_{\text{new}}(a_t \mid s_t) \propto \exp\left(Q^{\pi_{\text{old}}}(s_t, a_t)\right)$ in the policy improvement step. That is, the policy's update is proportional to the exponential distribution of the Q. Therefore, when the Q function is more uniformly distributed, the induced policies are prone to inconsistency (single-peaked policy distribution deviates from the center of the maximum Q).

Intuitively, alleviating the inconsistency in RL requires focusing on two aspects: (1) We should pay more attention to the states with more uniform action distribution during optimization; (2) Since the true action label is not available in advance, we need to reduce the probability of misdirection, i.e. closer value estimation to oracle.

3.3.3. The problem formulation and practical implementation

Let Q^* be the Q-function of the true action label, we give the optimization objective of inconsistency minimization,

$$\begin{aligned} \min_{\omega_k} \quad & \mathbb{E}_{d^{\pi_k}(s,a)} \left[\left| Q_k - Q^* \right| \right] \\ \text{s.t.} \quad & Q_k = \arg\min_{Q \in \mathcal{Q}} \mathbb{E}_{d^{\mu}} \left[\omega_k(s,a) \cdot \left(Q - \mathcal{B}^* Q_{k-1} \right)^2 (s,a) \right], \\ & \mathbb{E}_{d^{\mu}} \left[\omega_k(s,a) \right] = 1, \quad \omega_k(s,a) \geq 0, \end{aligned} \tag{13}$$

where $\pi_k(s) = \frac{\exp(Q_k(s,a))}{\sum_{a' \in \mathcal{A}} \exp(Q_k(s,a'))}$ is the policy corresponding to Q_k. Q_k is the estimate of Q-value after the Bellman update at iteration k. $\mathcal{Q}$ is the function space of Q-functions, d^{μ} is the data distribution of the replay

buffer and ω_k is the sample's weight. DisCor (Kumara et al., 2020) gives the upper bound:

$$\mathbb{E}_{d^{\pi_k}}\left[\left|Q_k - Q^*\right|\right] \leq \underbrace{\mathcal{H}\left(d^{\pi_k}\right)}_{(a)} \underbrace{-\log\left(\sum \exp\left(-\left|Q_k - Q^*\right|\right)\right)}_{(b)}. \tag{14}$$

Term (a) is the marginal state–action entropy of the policy π. DisCor bounds the $\mathcal{H}\left(d^{\pi_k}\right)$ by the entropy of the uniform distribution $\mathcal{H}\left(\mathcal{U}\right)$ for tractable. However, the inconsistency between the policy improvement and exploitation is indirectly increased by using that upper bound. That is, evenly-distributed action distributions deserve more attention than unevenly-distributed ones. To minimize the term (a), we draw inspiration from the work (Wang et al., 2021) in LDL which assigns higher weights to samples with larger information entropy to alleviate the inconsistency. Specifically, we give extra attention to the sample with larger entropy of action distribution to alleviate the inconsistency, i.e., reweighting samples $w.r.t.$ $\omega(s,a) := \mathcal{H}(\pi(\cdot|s))$.

Note that it is not reasonable to just introduce entropy weights for Eq. (13), because $\mathcal{B}^*Q_{k-1}$ is a bootstrapped target which may not be the ground truth label Q^*. Due to the accumulation of Bellman errors, Q_{k-1} has a gap from Q^*. Therefore, we need to minimize $\left|\mathcal{B}^*Q_{k-1} - Q^*\right|$. With an application of triangle inequality, we have:

$$\begin{aligned}\left|\mathcal{B}^*Q_{k-1} - Q^*\right| &= \left|\mathcal{B}^*Q_{k-1} - Q_k + Q_k - Q^*\right| \\ &\leq \left|Q_k - \mathcal{B}^*Q_{k-1}\right| + \left|Q_k - Q^*\right|\end{aligned} \tag{15}$$

That is, $\left|Q_k - Q^*\right|$ can serve as a surrogate. This result is corresponding to term (b), which aims to minimize Bellman error accumulation. In Eq. (15), the first term is the approximation error of the bootstrap error function, whose value is determined by the approximation algorithm and cannot be controlled. The second term is the distance between the Q estimate and the ground-truth. Liu et al. (2021) suggests that the value of $\left|Q_k - Q^*\right|$ is related to the "distance to the end". The bootstrap error will accumulate in the reverse direction with the trajectory. To solve it, we introduce a confidence weight (Auer, Cesa-Bianchi, & Fischer, 2002):

$$\omega(s,a) := \frac{\mathcal{T}(s,a)}{\mathcal{S}(s,a)}, \tag{16}$$

where $\mathcal{T}(s,a) = \sqrt{\ln t(s,a)}$, $t(s,a)$ is the step of sample (s,a) in every episode. Different with Liu et al. (2021), we introduce the $\mathcal{S}(s,a)$ which denotes the times the experience has been reused. Usually, the higher the number means the older the sample is, the lower the value of reuse. $\mathcal{S}(s,a)$ downweights those samples that have been reused many times. Thus, the inconsistency minimization weight is:

$$\omega^{\text{im}}(s,a) := \mathcal{H}(\pi(\cdot|s)) \cdot \frac{\mathcal{T}(s,a)}{\mathcal{S}(s,a)}. \tag{17}$$

3.3.4. The connection between inconsistency minimization and exploitation

From the perspective of exploration and exploitation, reducing cognitive inconsistency is often associated with a decrease in the agent's ability to explore. This initiative contradicts prior research and experience that emphasizes the importance of exploration. Increasing exploration can help minimize the risk of getting stuck in a local optimum. However, as discussed in Section 3.3.3, our approach to alleviating inconsistency requires accurate Q-value as a prerequisite. Our method is not an attempt to solely "exploit" but rather to reduce meaningless exploration.

Additionally, in Section 3.1, we illustrated the pathological concern of on-policiness prioritization in some cases and gave a correction strategy for it. An often neglected fact is the actual experience gathered depends on the agent's policy when interacting with the environment (Kumara et al., 2020). Therefore, we propose to consider possible future samples to be collected when optimizing the policy. We highlight that one could be more conducive if high-yield policies (i.e. self-imitating policies) can be enhanced to supply high-yield experiences.

Algorithm 1 The Cognitive Consistency (CoCo) framework

Initialize $Q_\theta(s,a)$, normal replay buffer $\mathcal{D}$, si-buffer $\mathcal{D}_{\text{si}}$, neural network κ_ψ, the highest return R^{MAX}, the reused times $\mathcal{S}(s,a)$.
for each environment step t **do**
  Collect new transition $\left(s_t, a_t, r_t, s_{t+1}, \mathcal{H}\left(\pi\left(\cdot \mid s_t\right)\right), t\left(s_t, a_t\right)\right)$ using π and add it to buffer $\mathcal{D}$.
  if Finish the current episode **then**
    Calculate the total return $R_\tau^\pi = \Sigma_{t=0}^T \gamma^t r(s_t, a_t)$ of the current episode $\tau = \left\{s_t, a_t\right\}_{t=0}^T$.
    if $R_\tau^\pi \geq R^{\text{MAX}}$ **then**
      Add samples $\left\{\left(s_t, a_t, r_t, s_{t+1}\right)\right\}_{t=1}^T$ gathered from the episode τ to si-buffer $\mathcal{D}_{\text{si}}$.
    end if
  end if
  if Reach the update interval **then**
    Obtain samples from $\mathcal{D}$ and $\mathcal{D}_{\text{si}}$.
    Update κ_ψ using: $L_\kappa(\psi) := \mathbb{E}_{\mathcal{D}}\left[f^*\left(f'\left(\kappa_\psi(s,a)\right)\right)\right] - \mathbb{E}_{\mathcal{D}_{\text{si}}}\left[f'\left(\kappa_\psi(s,a)\right)\right]$.
    Compute ω^{si} with the updated κ_ψ.
    Compute ω^{im} using Eq. (17).
    Minimize Bellman error for Q_θ weighted by $\omega = \omega^{\text{si}} \cdot \omega^{\text{im}}$:

$$L_Q(\theta; d^\mu, \omega) = \mathbb{E}_{d^\mu}[\omega(s,a)(Q_\theta(s,a) - \mathcal{B}^*Q_\theta(s,a))^2].$$

    Update π with base algorithm.
    Update the reused times $\mathcal{S}(s,a)$.
  end if
end for

On this basis, it is easy to think that the significance of inconsistency minimization lies in the ability to consolidate the policy and complement the samples it generated for exploitation.

Overall, the inconsistency minimization objective proposed in this section can effectively reduce inconsistencies and plays a crucial role in balancing the trade-off between exploration and exploitation, while also enhancing sample efficiency.

3.4. The cognitive consistency (CoCo) framework

3.4.1. Architectural overview

We first give the complete prioritization weight combined with the self-imitating distribution correction weight $\omega^{\text{si}}(s,a)$ and the inconsistency minimization weight $\omega^{\text{im}}(s,a)$:

$$\omega(s,a) := \frac{d^{\pi^{\text{si}}}(s,a)}{d^\mu(s,a)} \cdot \mathcal{H}(\pi(\cdot|s)) \cdot \frac{\mathcal{T}(s,a)}{\mathcal{S}(s,a)}. \tag{18}$$

We then present the Cognitive Consistency (CoCo) framework, which uses weight $\omega(s,a)$ to reweight the TD learning of value function. It is worth noting that CoCo works solely through a briefly reweighted uniformly sampled loss function, without the need for additional rewards (e.g., intrinsic rewards) or auxiliary losses (e.g., supervised losses) (Li, Gao, Yang, Xu, & Wu, 2022). This makes it more flexible when combined with other techniques. The overall structure of CoCo is shown in Fig. 3 and the pseudo-code is presented in Algorithm 1.

Intuitive Explanation. First, the samples generated from each episode are deposited into a buffer $\mathcal{D}$ according to the normal process of off-policy RL. Meanwhile, we calculate the total return R_τ^π of the current episode τ, and store the samples also into si-buffer $\mathcal{D}_{\text{si}}$ if R_τ^π greater than the highest total return R^{MAX}. Then, one batch is extracted from the buffer $\mathcal{D}$ and one from the si-buffer $\mathcal{D}_{\text{si}}$, while inputting into the neural network κ_ψ and updating its parameters. Next, we feed the batch drawn from the original buffer $\mathcal{D}$ into the updated κ_ψ and calculate the self-imitating distribution correction weight ω^{si}. Simultaneously, the inconsistency minimization weight ω^{im} of this batch is obtained by

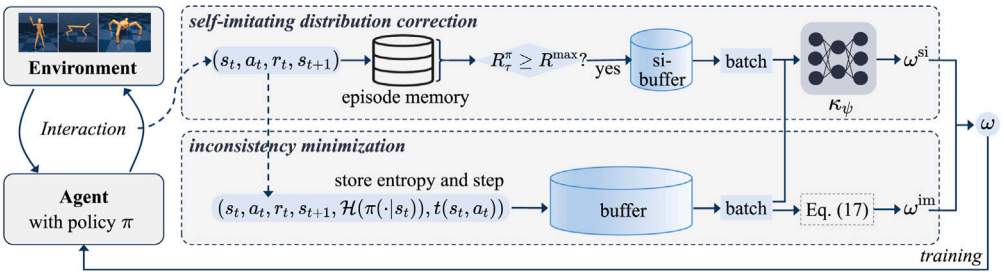
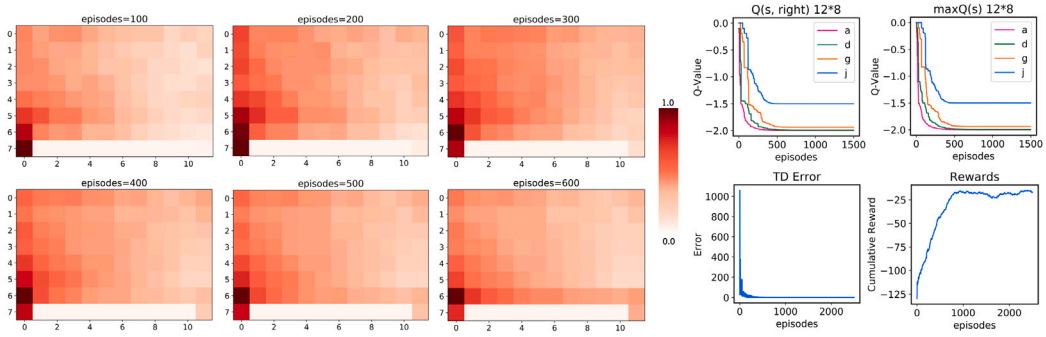


Fig. 3. Illustration of CoCo. CoCo consists of two components. In the self-imitating distribution correction component, the samples generated by the high-yield policy are stored as guide data to correct the sampling distribution. In the inconsistency minimization component, it records the action distribution entropy $\mathcal{H}(\pi(\cdot|s))$ and time steps $t(s,a)$ during sample collection, which are used to calculate the inconsistency minimization weight. Finally, the weightings of the two components are integrated to jointly guide the training.



(a) The visualized state visits. (b) The related values.

Fig. 4. Performance of CoCo on the Cliff task in Section 3.1.

Eq. (17). Finally, we integrate ω^{si} and ω^{im} to get the final weight ω, and use Eq. (1) to learn the value function for training agent.

3.4.2. Mutually beneficial analysis

The two components of CoCo are mutually beneficial. Intuitively, the *self-imitating distribution correction* captures the high-yield policies so that the *inconsistency minimization* does not act on the poor performing ones. Meanwhile, the inconsistency minimization consolidates the captured good policies, complementing the high-yield experiences and thus increasing their frequency of sampling. Such interactions accelerate learning efficiency. To verify our intuition, we validate the performance of CoCo on the modified Cliff task (Section 3.1), as shown in Fig. 4.

Fig. 4(a) shows the visualized state visits of CoCo on the Cliff task. CoCo has locked the optimal path at 600 episodes, which saves a third of the time compared to on-policiness. Furthermore, while CoCo does explore the low-yield regions (i.e., the upper left) due to the impact of exploration, it does not stray into those areas and only makes occasional visits. At 300 episodes, the agent has shifted the focus of the visit to the central region after visiting the optimal path. It is already roughly exploring only around the optimal path by about 500 episodes.

In Fig. 4(b), we observed that the convergence of the Q-value is consistent with that of the on-policiness exhibited in Fig. 1(d). However, the CoCo decreases more rapidly in the beginning, suggesting that updates at key states receive more attention. Besides, by examining the TD error curve, we found that the value rapidly decreases and maintains an extremely low value until convergence at around 500 episodes, indicating that the value function's update continues to focus on a narrow region. Finally, the CoCo exhibits convergence to smaller and more stable episode returns, indicating that the agent only explores a small area around the optimal path. We attribute this to the alleviation of inconsistency. The experimental results validate the mutually beneficial relationship between the two components of CoCo.

Additionally, the above conclusion can also be reflected in the following Lemma,

Lemma 3.5 (*Refer to Lemma. 5 in Liu et al., 2021*). *Let π^* be the optimal policy and π_k is the policy at iteration k, the performance discrepancy and the Q-value gap satisfy the following relationship:*

$$J\left(\pi^*\right) - J(\pi_k) \leq \frac{2}{1-\gamma}\left(\mathbb{E}_{d^{\pi_k,\pi^*}}\left|Q^*(s,a) - Q_k(s,a)\right| + 1\right). \tag{19}$$

where $d^{\pi_k,\pi^*}(s,a) = d^{\pi_k}(s)\frac{\pi_k(a|s)+\pi^*(a|s)}{2}$.

In Lemma 3.5, the bound is influenced by the discount factor γ. Nevertheless, we focus on the key relationship revealed by this lemma: the decrease of $\left|Q^*(s,a) - Q_k(s,a)\right|$ contributes to approximate to the optimal policy. While γ influences the bound's tightness, this relationship remains fundamental to our theoretical framework.

4. Experiments

In this section, we conduct experiments to evaluate the gains of CoCo in sample efficiency and performance. We first compare CoCo with related state-of-the-art on-policiness algorithms on Mujoco (Todorov, Erez, & Tassa, 2012) tasks. Meanwhile, we design ablation experiments to validate the components of CoCo and show that CoCo is robust to noisy reward. Then, we evaluate our methods on Atari games with discrete action spaces, especially demonstrating that CoCo is suitable for hard-exploration tasks. Detailed parameter settings, implementations, and code are available on GitHub[1] and Appendix B.

4.1. Performance on Mujoco tasks

To verify whether the proposed method CoCo can alleviate the inefficiency caused by on-policiness acting on low-yield regions and obtain a superior performance rapidly, we select the following methods as the compared baselines.

[1] https://github.com/DkING-lv6/CoCo.

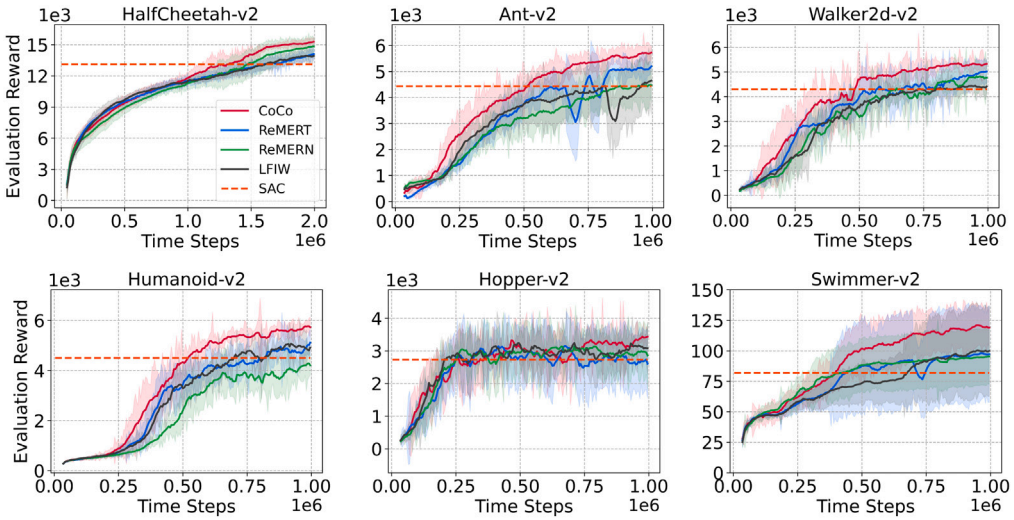


Fig. 5. Performance of CoCo (ours), ReMERT, ReMERN and LFIW combined with SAC on Mujoco tasks. The shaded region represents the standard deviation of the average evaluation over four trials with different random seeds.

- ReMERT (Liu et al., 2021) prioritizes samples with higher hindsight TD error, better on-policiness, and more accurate Q-value, which has demonstrated state-of-the-art performance on many tasks.
- ReMERN (Liu et al., 2021) emphasizes the prioritization of Bellman error estimation and outperforms the related method DisCor (Kumara et al., 2020).
- LFIW (Sinha et al., 2022) encourages small TD errors on the value function over frequently encountered states, whose performance serves as the benchmark for on-policiness.

We incorporate the above algorithms over the competitive algorithm SAC (Haarnoja et al., 2018). We use the same parameters as those used in these methods, including the total number of time steps, seeds, and learning rates. It is worth emphasizing that we do not directly intervene in the exploration strategy, and our approach is orthogonal to the design of exploration schemes. In contrast, we focus more on the "exploitation" phase; therefore, this paper does not directly compare with exploration-based methods.

Fig. 5 shows CoCo could have an excellent performance on most tasks except a comparable result on Hopper. There is not much difference of the $|Q_k - Q^*|$ between all the sampled state–action pairs since Hopper is a simple task. Therefore, prioritizing the samples fail to impact the overall performance improvement of the task significantly. It is derived from Kumara et al. (2020) and Liu et al. (2021) that both DisCor, ReMERT and ReMERN are still limited in terms of performance improvement or require more trials to achieve good performance (more than 2M or even 5M). It reflects that pursues more accurate value estimation to guarantee performance, which may rely on more interactions. However, CoCo demonstrates a surprising performance on these tasks. It achieves good results on five tasks earlier than the other methods, and even on Walker2d and Humanoid, only need about 0.5M interactions to achieve the 1M performance of others.

Since the standard deviations of final performance of CoCo overlaps with ReMERT on a couple of tasks, we provide a reliability analysis (Agarwal, Schwarzer, Castro, Courville, & Bellemare, 2021) of the results across all runs (on all the Mujoco tasks). We normalize our experimental data using the performance of SAC as a benchmark and provide the corresponding Interquartile Mean (IQM) results, which are shown in the Fig. 6. It validates that CoCo has the best performance improvement (the largest mean value) and is statistically significant.

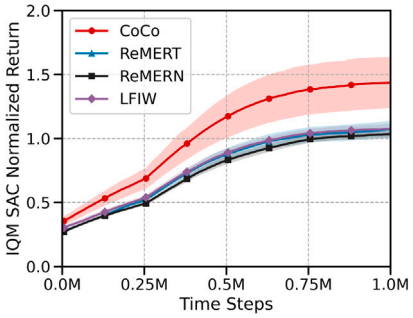


Fig. 6. The interquartile mean (IQM).

4.2. Ablation experiments

CoCo contains two components self-imitating distribution correction and inconsistency minimization that we denote separately as CoCo_sc and CoCo_im. Fig. 7 illustrates the results of the ablation study performed on these two components. CoCo_sc focuses on past high-yield samples to the extent that the trained policies are prone to cover only a narrow region of the state space. Therefore, it gradually converges to a sub-optimal result after showing some good trends in the beginning phase. CoCo_im aims at minimizing the inconsistency, which can reduce the generation of low-yield samples and the accumulation of Bellman errors. It is more stable while having good but not outstanding performance. From this, we need both ω^{si} to act as a guide and ω^{im} to ensure the accuracy of the update and the importance of the samples in the replay buffer. In addition, we demonstrate that the proposed inconsistency minimization component (which multiplies entropy by an additional term) is claimed to be better than DisCor's choice of entropy. We provide the results of the ablation experiments of CoCo_im with DisCor in Fig. 8. It can be seen that relying only on the inconsistent minimization weights, our method outperforms DisCor in most cases.

4.3. Performance with reward noise

CoCo relies on evaluating the total return to episodes, which can be affected by reward noise. To verify CoCo is robust to outliers, we modify the reward function $r(s, a)$ in the tasks to be equal to: $r'(s, a) =$

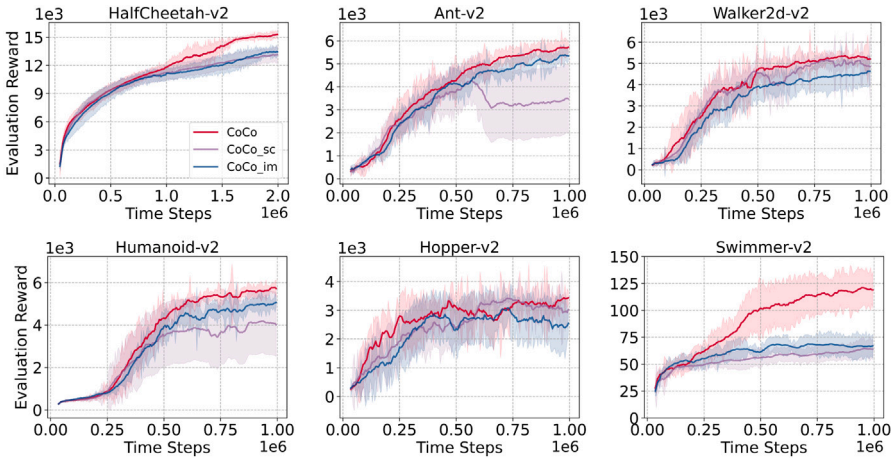


Fig. 7. Performance of CoCo, CoCo_sc, and CoCo_im combined with SAC.

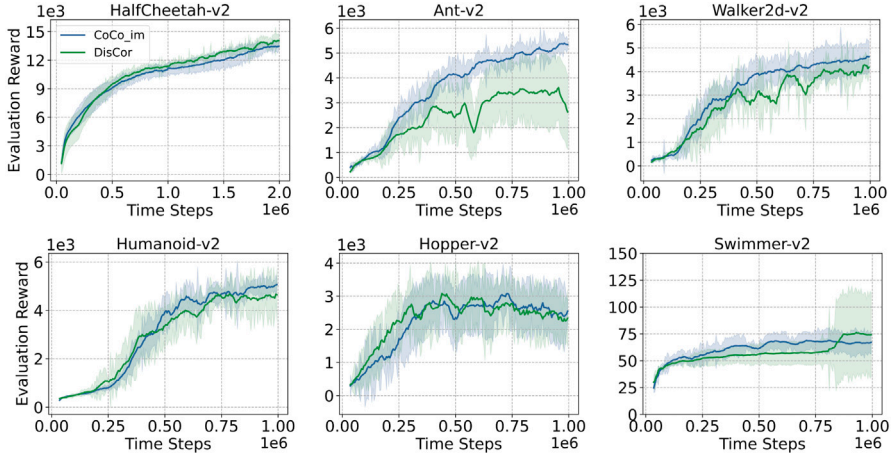


Fig. 8. Performance of CoCo_im, and DisCor combined with SAC.

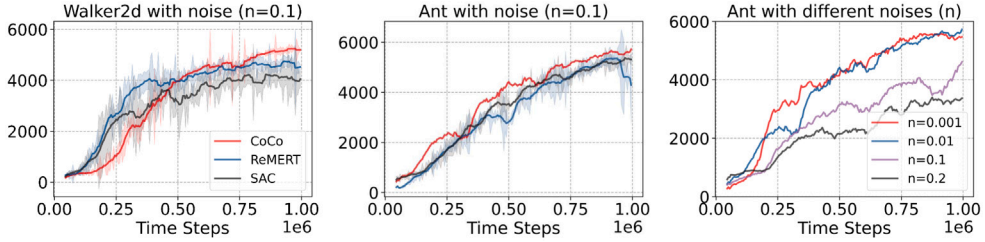


Fig. 9. The results of noisy reward tasks.

$r(s,a) + n \cdot z$, where $z \sim \mathcal{N}(0,1)$. We present the experimental results for the task involving reward noise in Fig. 9. CoCo is affected by the noise, but only during the initial stages of training. The reason may be that the agent may not have received significantly diverse incentives early in the training phase. This issue might lead CoCo to inadvertently add noisy samples to the si-buffer as guiding samples, as seen in Walker2d before 0.5M. However, the si-buffer is gradually updated to include better samples when the cumulative noise return has no impact on the better trajectory. Consequently, notable performance improvements are observed in Walker2d after 0.5M. We can conjecture that the cumulative reward noise typically affects the ranking of the total episode returns when n is large. This conjecture is supported by the performance of CoCo with different noises, as shown in Fig. 9 (right).

4.4. Performance on Atari games

We also demonstrate that our CoCo framework is competent for environments with discrete action space. Since the above methods in the previous section have no open-source code available for Atari experiments and considering the comparison with SIL (Oh et al., 2018), we choose Advantage Actor-Critic (A2C) (Mnih et al., 2016) as the baseline algorithm. We keep the same setting as SIL and run 5M steps (20M frames) of training on 20 Atari games. As shown by Table 1, CoCo outperforms SIL and A2C in almost all the tasks. To further evaluate our method, we measure improvement in percentage in score over the better of human and baseline agent scores (Wang et al., 2016):

$$\frac{\text{Score}_{\text{Agent}} - \text{Score}_{\text{Baseline}}}{\max\left\{\text{Score}_{\text{Human}}, \text{Score}_{\text{Baseline}}\right\} - \text{Score}_{\text{Random}}}, \tag{20}$$

Table 1
Performances on 20 Atari games after 5M steps of training (20M frames). *Imp.* means the improvement measured by (Wang et al., 2016).

Game	Random	Human	A2C	A2C+SIL		A2C+CoCo	
	score	score	score	score	Imp.	Score	Imp.
Alien	227.8	6875	739.7	1251.8	8%	**1767.2**	15%
Amidar	5.8	1676	189	248.9	4%	**365.8**	11%
Assault	222.4	1496	891.2	**958.3**	5%	933	3%
Asterix	210	8503	1869.4	2542.7	8%	**3472.1**	19%
Atlantis	12 850	29 028	45 672.3	53 189.2	23%	**63 765.5**	55%
BankHeist	14.2	734.4	**1147.8**	1031.2	−10%	898.2	−22%
BattleZone	2360	37 800	3151	8864	16%	**11 093.3**	22%
BeamRider	363.9	5775	750.8	1576.5	15%	**1824.7**	20%
Boxing	0.1	4.3	2.2	9.8	181%	**19.3**	407%
Breakout	1.7	31.8	25.2	51.5	87%	**59.6**	114%
Freeway	0	29.6	0	25.7	87%	**33**	111%
Gravitar	173	2672	67.3	276.8	8%	**871.2**	32%
Jamesbond	29	406.7	35.2	164.2	34%	**283.7**	66%
Kangaroo	52	3035	41.3	478	15%	**582.3**	18%
Montezuma's Revenge	0	4367	0	503.8	12%	**1896**	43%
MsPacman	307.3	15 693	1514.8	1878.1	2%	**2127.5**	4%
Qbert	163.9	13 455	2182.6	8411.2	47%	**14 843.2**	5%
SpaceInvaders	148	1652	249.7	486.5	16%	**755.7**	34%
UpNDown	533.4	9082	3641.2	10 126.2	76%	**15 733.6**	141%
Zaxxon	32.5	9173	124.2	5372.1	57%	**6469.7**	69%

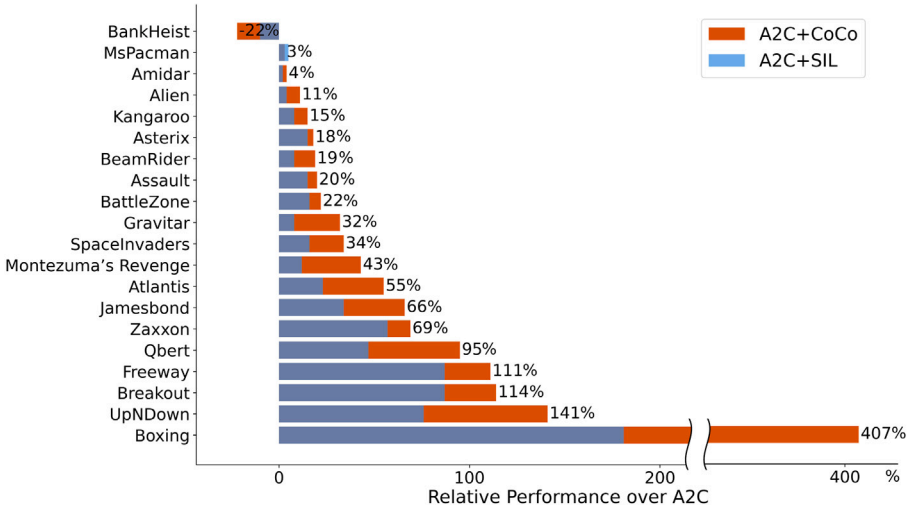


Fig. 10. The relative performance over A2C.

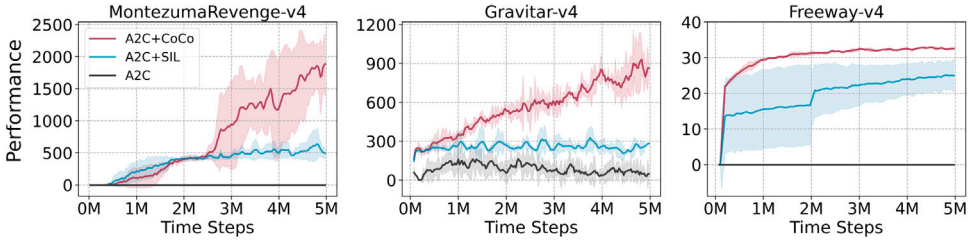


Fig. 11. The results of hard-exploration Atari games.

where $\text{Score}_{\text{Human}}$ and $\text{Score}_{\text{Baseline}}$ come from Mnih et al. (2015). The visualization result is shown in Fig. 10. We also record the time steps required for the relative algorithm to exceed human in Table 2, to show that CoCo has less training cost. It turns out that CoCo delivers significant efficiency gains. Even after removing a few tasks where only CoCo exceeds human, CoCo still consumes 35% fewer training steps than the relevant algorithm evaluation.

RL algorithms on hard-exploration tasks often lead to low-yield exploration. However, as demonstrated in Fig. 11, CoCo is capable of successfully handling these tasks. This is due to CoCo's ability to effectively leverage high-yield samples, which are crucial for researching hard-exploration tasks. As the enhanced policy approaches the next source of reward, it promotes deep exploration.

4.5. Further extension: the myopic issue of CoCo

We have noticed that our CoCo framework could be myopic as we prioritize cognitive conservatism. To illustrate, let us consider a grid world scenario where the optimal path incurs a significant negative

Table 2
Time steps (M) needed for the relative algorithm to exceed human. '–' indicates the score is below human-level even after 50M steps. SIL and CoCo represent A2C+SIL and A2C+CoCo. RED1 measures steps needed in percentage that CoCo reduced over A2C and RED2 measures that over SIL.

Game	A2C	SIL	CoCo	RED1	RED2	Mean
Alien	–	–	–	–	–	–
Amidar	–	–	–	–	–	–
Assault	32.2	13.9	16.8	48%	−21%	13%
Asterix	33.8	24.4	10.9	68%	55%	62%
Atlantis	4.1	3.6	2.7	34%	25%	30%
BankHeist	4.8	4.1	4.3	10%	−5%	3%
BattleZone	–	–	–	–	–	–
BeamRider	–	–	–	–	–	–
Boxing	5.6	4.5	3	46%	33%	40%
Breakout	6.3	4.4	3.9	38%	11%	25%
Freeway	–	6.4	0.8	–	88%	–
Gravitar	–	–	–	–	–	–
Jamesbond	–	–	–	–	–	–
Kangaroo	–	–	34.7	–	–	–
Montezuma's Revenge	–	–	–	–	–	–
MsPacman	–	–	–	–	–	–
Qbert	14.8	7.3	4.7	68%	36%	52%
SpaceInvaders	37.3	25.2	18.8	50%	25%	37%
UpNDown	9.8	4.3	2.6	73%	40%	57%
Zaxxon	–	–	–	–	–	–
Normed Mean	16.5	10.2	7.5	48%	29%	35%
Normed Median	9.8	4.5	4.3	48%	29%	37%

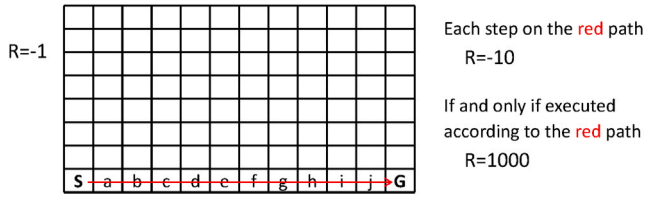


Fig. 12. The diagram.

penalty at every time step, except for the final step when the optimal action results in an exceptionally large reward. At each time step, the agent has the option to move forward or terminate the environment. In this case, our approach would encourage the agent to converge on a sub-optimal path. In order to assess its performance, we make the following changes to the Cliff task:

1. The cliff area is canceled, and the corresponding optimal path becomes the last line in the Grid, such as the red path in Fig. 12.
2. The optimal trajectory gets a large negative reward −10 on each time step.
3. If and only if the agent executes according to the red route can get an extremely large reward 1000.

The results presented in Fig. 13 demonstrate that our method can be myopic in certain scenarios. Specifically, in the modified Cliff task, our method ultimately converges to a sub-optimal solution (the path in the penultimate row of the grid). This can be observed from the reward value, as the policy that eventually converges cannot achieve the optimal reward of 1000. Additionally, we tested the on-policiness method and found that it too produces sub-optimal results. However, our method appears to be more robust because the right actions with larger negative rewards have smaller Q values.

For the myopic phenomenon described above, we attribute it to the somewhat harsh conditions for reaching the optimal path. It is extremely difficult to explore the red path with a large negative reward. Therefore, we further simplified the Cliff task by reducing its scope to 6*4, and the experimental results are shown in Fig. 14. In the task settings, we specifically increased the exploration rate and reduced the temperature coefficient. It is clear that neither our method nor the on-policiness method ever reach the ideal policy. The few high-quality samples obtained by merely three explorations of the optimal path cannot be used effectively.

The aforementioned myopic issue is not unique to our work but also exists in existing on-policiness work. However, it is possible that combining our work with some special exploration techniques can be a positive contribution.

5. Conclusion and future work

We propose to rethink the trade-off between exploration and exploitation from a novel perspective of cognitive consistency and introduce a framework termed CoCo. The core of CoCo lies in conducting pessimistic exploration and optimistic exploitation under reasonable premises. We highlight that CoCo can enhance sample efficiency without compromising performance. In CoCo, we first use a self-imitating distribution correction approach to pursue cognition-oriented optimistic exploitation. Then, we innovatively introduce an inconsistency minimization objective inspired by LDL to achieve pessimistic exploration. Extensive experiments validate our framework and its properties, demonstrating that rational utilization of cognitive consistency can substantially improve sample efficiency and performance of standard off-policy RL methods.

In future work, an exciting direction would be to generalize our framework to model-based RL and offline RL. Additionally, the inconsistency minimization objective investigates the RL problem from the perspective of label distribution learning, which could inspire future investigation.

CRediT authorship contribution statement

Da Wang: Writing – review & editing. **Wei Wei:** Writing – review & editing. **Lin Li:** Writing – review & editing. **Xin Wang:** Writing – review & editing. **Jiye Liang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2020AAA0106100), the National Natural Science Foundation of China (62276160), the Natural Science Foundation of Shanxi Province, China (202203021211294), and the Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shanxi Province (20240002).

Appendix A. Related work

A.1. Sample efficiency

Sample efficiency remains a key challenge in large-scale practical applications. Training a good policy may require millions (or even billions) of environment steps, and this problem is exacerbated when the data collection is expensive. Off-policy RL provides better sample efficiency than its on-policy counterparts, owing to its ability to learn from data distributions that are not constrained by the current policy. Off-policy RL provides better sample efficiency (Riedmiller, Springenberg, Hafner, & Heess, 2022; Schwarzer et al., 2021; Yu, 2018) than on-policy counterparts, owing to its ability to learn from data distributions not constrained by the current policy. It has shown to be extremely valuable in robotics applications and crucial to the advancement of offline RL (Riedmiller et al., 2022), especially in light of the rise in prominence of off-policy actor-critic algorithms (Haarnoja et al., 2018; Heess et al., 2015; Lillicrap et al., 2021).

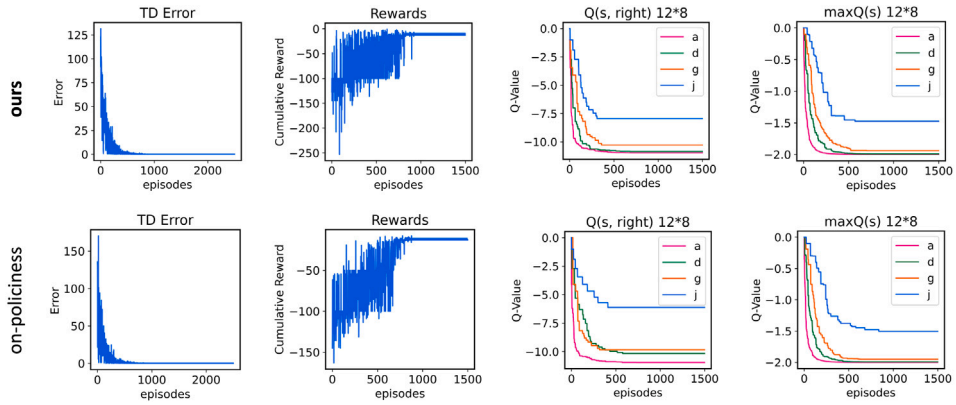


Fig. 13. A modified 12*8 Cliff example with a large negative reward on each time step for taking the optimal action.

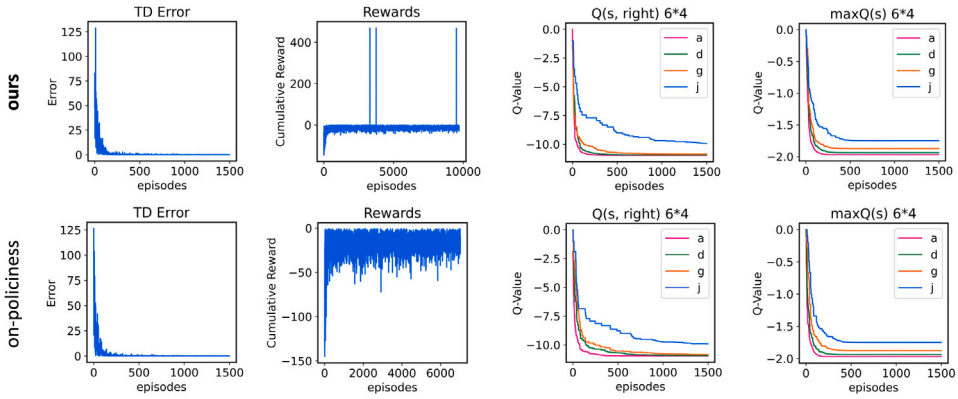


Fig. 14. A modified 6*4 Cliff example with a large negative reward on each time step for taking the optimal action.

A.2. Cognitive consistency

Human beings strive to maintain balance and harmony within their internal cognitive systems, which is a process commonly referred to as maintaining cognitive consistency (Festinger, 1962). In social psychology, cognitive consistency refers to the preservation of existing knowledge structures, such as percepts, schemata (categories), memories, and propositions (Greenwald, 1980). Together with the abilities to imitate (Arora & Doshi, 2021), compositionally generalize (Liu & Frank, 2022), and make inferences (Olson, Khanna, Neal, Li, & Wong, 2021), cognitive consistency is a crucial factor that enables humans to acquire effective policies quickly. It can enhance memory retention and information processing, improving the efficiency and quality of learning. Furthermore, it can enhance cognitive stability and predictability, reducing conflict and uncertainty, thus easing the burden during the learning process. For instance, when the library needs to add a new batch of books, it may be more efficient to maintain consistency with the already established cataloging scheme rather than allocating librarian effort to revise it every time by recataloging and reshelving the existing collection (Greenwald, 1980).

Previous research has highlighted the importance of maintaining consistent cognition among agents in multi-agent systems to achieve effective system-level cooperation (Bear, Kagan, & Rand, 2017; Corgnet, Espín, & Hernán-González, 2015; Oroojlooy & Hajinezhad, 2022). NCC (Mao et al., 2020) introduces Neighborhood Cognitive Consistency into multi-agent RL by representation alignment between neighbors, indicating that maintaining neighborhood cognitive consistency is usually sufficient to ensure system-level cooperation. In this work, we take the first step towards introducing cognitive consistency in standard off-policy RL for single agents (Hessel et al., 2018; Mnih et al., 2015). We propose that cognitive consistency is expressed in off-policy RL as the unity of knowledge and behavior. This consistency can manifest in various ways, such as a preference for certain actions or a tendency to avoid risky decisions. Another related concept is the *exploration–exploitation trade-off*, which refers to the balance between trying new behaviors (exploration) and utilizing current knowledge (exploitation) to maximize cumulative rewards. The exploitation in off-policy RL is expressed as exploiting past experiences (Oh et al., 2018). Existing works use experience replay which frequently prioritizes or reweights experiences with criteria such as TD error (Schaul et al., 2016), corrective feedback (Kumara et al., 2020; Lee et al., 2021) and on-policiness (Liu et al., 2021; Novati & Koumoutsakos, 2019; Sinha et al., 2022; Sun et al., 2020; Wang et al., 2019). Another feasible approach is controlling the data collection, but the question of how this data is collected has been vastly understudied (Kumara et al., 2020). One promising direction is to modify exploration policies to manage experience collecting. Existing approaches concentrate on designing various intrinsic rewards to encourage more thorough exploration (Burda, Edwards, Storkey, & Klimov, 2019; Ecoffet et al., 2021; Han & Sung, 2021; Mavor-Parker et al., 2022; Pathak, Agrawal, Efros, & Darrell, 2017; Yuan et al., 2022; Zhang et al., 2021). Other works (Andrychowicz et al., 2017; Florensa, Held, Geng, & Abbeel, 2018) use generative techniques to complement valuable learning signals. These initiatives have had great success in sparse reward settings.

A.3. Experience replay

Riedmiller et al. (2022) state that sample-efficient RL goes through three phases: pure online RL, RL with a replay buffer, and finally offline RL. It is well known that pure online RL is inefficient in most scenarios because each data point is considered only once. In recent years offline RL (entirely without interaction) (Gulcehre et al., 2020; Lange, Gabel,

& Riedmiller, 2012; Li et al., 2022; Siegel et al., 2020) has attracted more attention, but there are still numerous obstacles to achieving widespread use in real-world scenes. In contrast, learning with a replay buffer, a fundamental component of off-policy RL, remains a long-term concern (Kumara et al., 2020; Liu et al., 2021; Sinha et al., 2022). This phase has shown to be extremely valuable in robotics applications and crucial to the advancement of offline RL, especially in light of the rise in prominence of off-policy actor-critic algorithms (Haarnoja et al., 2018; Heess et al., 2015; Lillicrap et al., 2021).

Prioritized Experience Replay (PER) (Schaul et al., 2016) considers samples with high TD error to be more important. Importance sampling is needed to correct for bias since the method of sampling by prioritization changes the data distribution. It introduces variance, although it ensures that expectations are unbiased (Liu, Li, Tang, & Zhou, 2018; Schlegel, Chung, Graves, Qian, & White, 2019). Loss-Adjusted Prioritized (LAP) (Fujimoto, Meger, & Precup, 2020) points out that some benefits of prioritized experience replay come from the change in the expected gradient rather than the prioritization itself. Thus, the design of prioritization sampling methods should not be considered in isolation from the loss function, and a uniform sampling method with the correct loss function can be an alternative to non-uniform sampling.

Likelihood-Free Importance Weighting (LFIW) (Sinha et al., 2022) argues that the prioritized experience replay of TD learning can be considered as choosing a favorable prioritized distribution (Nachum, Chow, Dai, & Li, 2019). It encourages small TD errors on the value function over frequently encountered states. Several similar works (Novati & Koumoutsakos, 2019; Sun et al., 2020; Wang et al., 2019) also emphasize the importance of recent samples for training the current policy (i.e., more on-policiness).

Distribution Correction (DisCor) (Kumara et al., 2020) demonstrates that bootstrapping based Q-learning algorithms do not benefit well from the "corrective feedback". It reduces the weight of samples for which the target Q-value estimate has a high cumulative error with Q^*. Inspired by this, SUNRISE (Lee et al., 2021) reweights the target Q-value based on uncertainty estimates by using the variance of the Q-ensemble.

Regret Minimization Experience Replay Using Temporal Structure (ReMERT) (Liu et al., 2021) indicates that previous prioritization criteria are mostly heuristically designed, which can be sub-optimal in some cases due to the mismatch with the RL objective. It suggests that the optimal sample prioritization strategy should satisfy higher hindsight TD error, better on-policiness, and more accurate Q-value. Compared with DisCor, the better on-policiness criterion is the key to policy correction.

To sum up, DisCor and REMERT advise to reduce the cumulative error of Q estimates, which is similar to part of our view. In addition, LFIW and REMERT indicate that prioritization or reweighting of samples with on-policiness can yield significant performance improvements. However, this type of method is hampered by the behavior policy. It may be inefficient when the agent pays excessive attention to the low-yield region exploration. In this work, we analyze the pathological concerns associated with the on-policiness priority criterion and correct it using the distribution of authoritative policies.

A.4. Self-imitation learning

Imitation learning (Abbeel & Ng, 2004; Arora & Doshi, 2021; Ho & Ermon, 2016; Torabi, Warnell, & Stone, 2018) learns a good policy by mimicking expert demonstrations. Self-imitation learning (SIL) (Oh et al., 2018) learns to reproduce past good decisions over self-generated experiences without external demonstrations. It is mainly achieved by using the supervised learning (SL) objective as an auxiliary loss and optimizing it jointly with the standard RL objective (Li et al., 2022). In our work, we introduce a novel self-imitating distribution correction approach that sets itself apart from SIL and similar algorithms in the following ways:

Table B.3
The symbols and abbreviations.

Symbol	Description	Symbol	Description
$\mathcal{S}$	State space	s	State
$\mathcal{A}$	Action space	a	Action
$\mathcal{P}$	Transition function	r	Reward function
γ	Discount factor	p_0	Distribution of the initial state
J	Cumulative rewards	t	Time steps
π	Policy	d^π	Discounted stationary state distribution
μ	Behavior policy	ω	Sample weight
$\mathcal{B}$	Bellman operator	$\mathcal{B}^*$	Bellman optimal operator
V^π	State value function	Q^π	State–action value function
$\mathcal{H}$	Entropy	τ	Trajectory
$R_{\max}$	Upper bound of the reward function	R^{MAX}	The highest return
ω^{si}	self-imitating weight	ω^{im}	inconsistency minimization weight
$\mathcal{D}$	Conventional buffer	$\mathcal{D}_{\text{si}}$	Smaller size buffer

- SIL imitates good decisions for each state individually, which is heuristic and probably leads to sub-optimal results. Our approach uses the ultimate goal of RL as a metric (Liu et al., 2021) to directly mimic the complete high-yield policy.
- SIL uses the difference between the observed return and the estimated value as a reward bonus, then utilizes PER (Schaul et al., 2016) to sample experiences from the replay buffer. It suffers from stale returns (Ferret, Pietquin, & Geist, 2021) and introduces bias. Our approach does not require reward setting and always guarantees the contribution of self-imitating experiences by storing them in an additional buffer. Besides, we reduce the bias by using a brief reweighted uniformly sampled loss function (Fujimoto et al., 2020).
- PhAsic self-Imitative Reduction (PAIR) (Li et al., 2022) points out that optimizing the mixed objective of RL and SL with SIL can be brittle and requires substantial parameter tuning. PAIR relies on self-generated samples as supervised signals for the offline SL phase, which does not require optimization of a mixed objective. Our approach shares some similarities with PAIR, but with the distinction that we use self-imitating samples as guidance to correct the data distribution over the replay buffer, without addressing offline RL.

Appendix B. Details

B.1. Symbols and abbreviations

See Table B.3.

B.2. Detailed parameter settings

The detailed parameter settings are listed in Table B.4. The algorithms ReMERT, ReMERN, LFIW, and CoCo (ours) all contain the full parameters of *Agent* and *SAC*. ReMERN introduces an error network to calculate the cumulative Bellman error with a learning rate of 0.0003 and a hidden network of [256, 256, 256]. In addition, ReMERN maintains a moving average of the temperatures initialized as 10.0 to perform the weighting.

The replay buffer size $|\mathcal{D}_{\text{f}}|$ of LFIW affects the number of experiences we treat as "on-policiness". According to LFIW's previous experience, the performance is relatively stable for $|\mathcal{D}_{\text{f}}| = 1 \times 10^5$. The hidden network sizes of κ_ψ are [128, 128], and the temperature hyperparameter T for self-normalization to the importance weights is 7.5. The normalization is:

$$\tilde{\kappa}_\psi(s,a) := \frac{\kappa_\psi(s,a)^{1/T}}{\mathbb{E}_{\mathcal{D}_{\text{s}}}\left[\kappa_\psi(s,a)^{1/T}\right]}.$$

Table B.4
Hyper-parameters for continuous control tasks.

	Hyper-parameters	Value
Agent	Training steps	Chosen from {0.5M, 1M, 1.5M, 2M}
	Buffer size	1×10^6
	Batch size	256
	Evaluation interval	5000
	Update interval	1
	Random seed	10, 100, 1000, 10000
SAC	γ	0.99
	Init α	1.0
	Actor learning rate	0.0003
	Critic learning rate	0.0003
	α learning rate	0.0003
	Hidden network sizes	[256, 256]
ReMERN	Error learning rate	0.0003
	Error hidden network sizes	[256, 256, 256]
	Init temperature τ	10.0
LFIW	Buffer size $\lvert D \rvert$	1×10^6
	Buffer size $\lvert D_{\mathrm{f}} \rvert$	1×10^5
	κ_ψ hidden network sizes	[128, 128]
	Temperature T	7.5
CoCo(ours)	si-buffer size $\lvert D_{\mathrm{si}} \rvert$	2×10^5
	Init R^{MAX}	-1000

ReMERT, ReMERN, and CoCo maintain uniform parameters with LFIW in calculating the likelihood-free importance weight. The difference is that CoCo sets the size of si-buffer to $\lvert D_{\mathrm{si}} \rvert = 2{\times}10^5$. It intends to prevent overfitting and catastrophic forgetting due to a lack of diversity.

B.3. Implementation details

Baselines. The source codes of all the above algorithms are provided by ReMER. [2] Our method CoCo also alters based on this and adds only a dozen lines of code. Some implementation details about CoCo are as follows. **Compute entropy** $\mathcal{H}(\pi(\cdot|s))$. For the algorithms in discrete action spaces, the entropy is calculated by $\mathcal{H}(y) = -\sum_j y_j \log y_j$. However, this study only emphasizes continuous control tasks. Thus, we use the differential entropy of Gaussian distribution:

$$\mathcal{H}\left[\mathcal{N}\left(\mu, \sigma^2\right)\right] = \frac{1}{2} \log 2\pi e \sigma^2.$$

Note that the result of the differential entropy may have negative values. We normalize them so that they can be used as sample weights:

$$\mathcal{H}(\pi(\cdot|s)) = \frac{\mathcal{H}(\pi(\cdot|s)) - \min \mathcal{H}(\pi(\cdot|s))}{\max \mathcal{H}(\pi(\cdot|s)) - \min \mathcal{H}(\pi(\cdot|s))}.$$

Compute confidence weight $\omega(s,a) := \frac{\mathcal{T}(s,a)}{\mathcal{S}(s,a)}$. To ensure that the $\mathcal{T}(s,a)$ and $\mathcal{S}(s,a)$ have the same magnitude, we compute the confidence weight using the following formula instead:

$$\omega(s,a) := \frac{\exp(\lambda^{t(s,a)+\mathcal{S}(s,a)})}{\mathrm{e}-1},$$

where $\mathcal{T}(s,a) = \sqrt{\ln t(s,a)}$, $t(s,a)$ is the step of sample (s,a) in every episode. $\lambda = 0.996$ and e is the natural logarithm.

Data availability

Data will be made available on request.

[2] https://github.com/AIDefender/MyDiscor.

References

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st international conference on machine learning* (p. 1).

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., & Bellemare, M. (2021). Deep reinforcement learning at the edge of the statistical precipice. In *Advances in neural information processing systems* (pp. 29304–29320).

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., et al. (2017). Hindsight experience replay. In *Advances in neural information processing systems* (pp. 5055–5065).

Arora, S., & Doshi, P. (2021). A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, *297*, Article 103500.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*(2–3), 235–256.

Bear, A., Kagan, A., & Rand, D. G. (2017). Co-evolution of cooperation and cognition: the impact of imperfect deliberation and context-sensitive intuition. *Proceedings of the Royal Society B: Biological Sciences*, *284*(1851), Article 20162326.

Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2019). Exploration by random network distillation. In *International conference on learning representations*.

Corgnet, B., Espín, A. M., & Hernán-González, R. (2015). The cognitive basis of social behavior: cognitive reflection overrides antisocial but not always prosocial motives. *Frontiers in Behavioral Neuroscience*, *9*, 287.

Csiszár, I. (1964). Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, *8*, 85–108.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., & Clune, J. (2021). First return, then explore. *Nature*, *590*(7847), 580–586.

Ferret, J., Pietquin, O., & Geist, M. (2021). Self-imitation advantage learning. In *AAMAS 2021-20th international conference on autonomous agents and multiagent systems*.

Festinger, L. (1962). *A theory of cognitive dissonance*: *vol. 2*, Stanford University Press.

Florensa, C., Held, D., Geng, X., & Abbeel, P. (2018). Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th international conference on machine learning* (pp. 1515–1528).

Fujimoto, S., Meger, D., & Precup, D. (2020). An equivalence between loss functions and non-uniform sampling in experience replay. In *Advances in neural information processing systems* (pp. 14219–14230).

Geng, X. (2016). Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, *28*(7), 1734–1748.

Greenwald, A. G. (1980). The totalitarian ego: Fabrication and revision of personal history. *American Psychologist*, *35*(7), 603.

Gulcehre, C., Wang, Z., Novikov, A., Paine, T., Gómez, S., Zolna, K., et al. (2020). Rl unplugged: A suite of benchmarks for offline reinforcement learning. In *Advances in neural information processing systems* (pp. 7248–7259).

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th international conference on machine learning* (pp. 1861–1870).

Han, S., & Sung, Y. (2021). A max-min entropy framework for reinforcement learning. In *Advances in neural information processing systems* (pp. 25732–25745).

Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., & Tassa, Y. (2015). Learning continuous control policies by stochastic value gradients. In *Advances in neural information processing systems* (pp. 2944–2952).

Hessel, M., Modayil, J., Hasselt, H. V., Schaul, T., Ostrovski, G., Dabney, W., et al. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the 32nd AAAI conference on artificial intelligence* (pp. 3215–3222).

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in neural information processing systems* (pp. 4565–4573).

Hsu, K.-C., Ren, A. Z., Nguyen, D. P., Majumdar, A., & Fisac, J. F. (2023). Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees. *Artificial Intelligence*, *314*, Article 103811.

Kumara, A., Gupta, A., & Levine, S. (2020). DisCor: Corrective feedback in reinforcement learning via distribution correction. In *Advances in neural information processing systems* (pp. 18560–18572).

Lange, S., Gabel, T., & Riedmiller, M. (2012). Batch reinforcement learning. *Reinforcement Learning: State-of-the-Art*, 45–73.

Lee, K., Laskin, M., Srinivas, A., & Abbeel, P. (2021). SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. In *Proceedings of the 38th international conference on machine learning* (pp. 6131–6141).

Leottau, D. L., del Solar, J. R., & Babuška, R. (2018). Decentralized reinforcement learning of robot behaviors. *Artificial Intelligence*, *256*, 130–159.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, *17*(1), 1334–1373.

Li, Y., Gao, T., Yang, J., Xu, H., & Wu, Y. (2022). Phasic self-imitative reduction for sparse-reward goal-conditioned reinforcement learning. In *Proceedings of the 39th international conference on machine learning* (pp. 12765–12781).

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2021). Continuous control with deep reinforcement learning. In *International conference on learning representations*.

Lin, L. J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, *8*(3–4), 293–321.

Liu, R. G., & Frank, M. J. (2022). Hierarchical clustering optimizes the tradeoff between compositionality and expressivity of task structures for flexible reinforcement learning. *Artificial Intelligence*, *312*, Article 103770.

Liu, Q., Li, L., Tang, Z., & Zhou, D. (2018). Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in neural information processing systems* (pp. 5356–5366).

Liu, Z., Lu, M., Xiong, W., Zhong, H., Hu, H., Zhang, S., et al. (2023). Maximize to explore: One objective function fusing estimation, planning, and exploration. In *Advances in neural information processing systems*.

Liu, X., Xue, Z., Pang, J., Jiang, S., Xu, F., & Yu, Y. (2021). Regret minimization experience replay in off-policy reinforcement learning. In *Advances in neural information processing systems* (pp. 17604–17615).

Mao, H., Liu, W., Hao, J., Luo, J., Li, D., Zhang, Z., et al. (2020). Neighborhood cognition consistent multi-agent reinforcement learning. *vol. 34*, In *Proceedings of the 34th AAAI conference on artificial intelligence* (pp. 7219–7226). 05.

Mavor-Parker, A., Young, K., Barry, C., & Griffin, L. (2022). How to stay curious while avoiding noisy tvs using aleatoric uncertainty estimation. In *International conference on machine learning* (pp. 15220–15240).

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd international conference on machine learning* (pp. 1928–1937).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Nachum, O., Chow, Y., Dai, B., & Li, L. (2019). Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in neural information processing systems* (pp. 2315–2325).

Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *56*(11), 5847–5861.

Novati, G., & Koumoutsakos, P. (2019). Remember and forget for experience replay. In *Proceedings of the 36th international conference on machine learning* (pp. 4851–4860).

Oh, J., Guo, Y., Singh, S., & Lee, H. (2018). Self-imitation learning. In *Proceedings of the 35th international conference on machine learning* (pp. 3878–3887).

Olson, M. L., Khanna, R., Neal, L., Li, F., & Wong, W.-K. (2021). Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence*, *295*, Article 103455.

Oroojlooy, A., & Hajinezhad, D. (2022). A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 1–46.

Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th international conference on machine learning* (pp. 2778–2787).

Riedmiller, M., Springenberg, J. T., Hafner, R., & Heess, N. (2022). Collect & infer-a fresh look at data-efficient reinforcement learning. In *Conference on robot learning* (pp. 1736–1744).

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016). Prioritized experience replay. In *International conference on learning representations*.

Schlegel, M., Chung, W., Graves, D., Qian, J., & White, M. (2019). Importance resampling for off-policy prediction. In *Advances in neural information processing systems* (pp. 1797–1807).

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd international conference on machine learning* (pp. 1889–1897).

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., & Bachman, P. (2021). Data-efficient reinforcement learning with self-predictive representations. In *International conference on learning representations*.

Siegel, N., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., et al. (2020). Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *International conference on learning representations*.

Sinha, S., Song, J., Garg, A., & Ermon, S. (2022). Experience replay with likelihood-free importance weights. In *Proceedings of the fourth annual learning for dynamics and control conference* (pp. 110–123).

Sun, H., Han, L., Yang, R., Ma, X., Guo, J., & Zhou, B. (2022). Exploit reward shifting in value-based deep-rl: Optimistic curiosity-based exploration and conservative exploitation via linear reward shaping. In *Advances in neural information processing systems* (pp. 37719–37734).

Sun, P., Zhou, W., & Li, H. (2020). Attentive experience replay. In *Proceedings of the 34th AAAI conference on artificial intelligence* (pp. 5900–5907).

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Todorov, E., Erez, T., & Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *Proceedings of the 24th international conference on intelligent robots and systems* (pp. 5026–5033).

Torabi, F., Warnell, G., & Stone, P. (2018). Behavioral cloning from observation. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 4950–4957).

Wang, J., Geng, X., & Xue, H. (2021). Re-weighting large margin label distribution learning for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd international conference on machine learning* (pp. 1995–2003).

Wang, C., Wu, Y., Vuong, Q., & Ross, K. (2019). Striving for simplicity and performance in off-policy DRL: Output normalization and non-uniform sampling. In *Proceedings of the 36th international conference on machine learning* (pp. 10070–10080).

Wei, W., Wang, D., Li, L., & Liang, J. (2024). Re-attentive experience replay in off-policy reinforcement learning. *Machine Learning*, *113*(5), 2327–2349.

Xu, T., Li, Z., & Yu, Y. (2021). Error bounds of imitating policies and environments for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(10), 6968–6980.

Yang, J., Zhao, Q., Wang, H., Huang, Y., Song, Z., & Fang, M. (2023). Optimistic and pessimistic actor in RL: Decoupling exploration and utilization. arXiv:2312.15965.

Ye, W., Liu, S., Kurutach, T., Abbeel, P., & Gao, Y. (2021). Mastering atari games with limited data. In *Advances in neural information processing systems* (pp. 25476–25488).

Yu, Y. (2018). Towards sample efficient reinforcement learning. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 5739–5743).

Yuan, M., Pun, M.-O., & Wang, D. (2022). Rényi state entropy maximization for exploration acceleration in reinforcement learning. *Artificial Intelligence*, *1*(1), 1–11.

Zhang, T., Rashidinejad, P., Jiao, J., Tian, Y., Gonzalez, J. E., & Russell, S. (2021). Made: Exploration via maximizing deviation from explored regions. In *Advances in neural information processing systems* (pp. 9663–9680).