Reexamining the Exploration–Exploitation Dilemma from an Entropy-Driven Perspective

Renye Yan[1], **Yaozhong Gan**[2*], **Jikang Cheng**[1], **Yi Sun**[2], **Zongwei Wang**[1], **Ling Liang**[1*] and **Yimao Cai**[1*]
[1]Peking University
[2]Tsinghua University
victory@stu.pku.edu.cn, {lingliang,caiyimao}@pku.edu.cn

Abstract

Achieving an optimal balance between exploration and exploitation remains a fundamental challenge in reinforcement learning. This work revisits the exploration-exploitation dilemma through the lens of entropy, offering a novel perspective on this enduring problem. It establishes a theoretical connection between policy entropy and exploratory behavior, using entropy as a rational measure to quantify the exploration-exploitation trade-off. Theoretical analyses demonstrate that a modified Bellman equation, augmented with a novelty-seeking term, ensures appropriate entropy adjustment and guarantees its globally monotonic decay. The derived policy optimization process inherently accommodates all three regimes of the exploration-exploitation spectrum, enabling a principled transition from exploration to exploitation. Building on these theoretical insights, this work introduces AdaZero, an adaptive deep architecture that dynamically balances exploration and exploitation. Extensive empirical evaluations highlight AdaZero's robust performance and validate the theory's feasibility.

1 Introduction

In reinforcement learning (RL) [Zhao *et al.*, 2022; Gan *et al.*, 2024; Yan *et al.*, 2026], the two fundamental components, i.e., exploration and exploitation, are simultaneously opposing and complementary. Too much exploration reduces learning efficiency, while excessive exploitation of uncertain knowledge hinders long-term reward maximization, leading to suboptimal policies [Tokic, 2010]. How to achieve an optimal balance of exploration and exploitation remains a significant challenge, known as the exploration-exploitation dilemma [Ahmed *et al.*, 2019; Schulman *et al.*, 2015, 2017; Bellemare *et al.*, 2016].

Methods based on policy and rewards have been proposed to address exploration-exploitation trade-offs in RL. For example, ϵ-greedy and its variations [Sutton and Barto, 1999; Tokic, 2010; Mnih *et al.*, 2015] randomly select exploratory and exploitative actions, which is simple to implement and broadly effective [Taiga *et al.*, 2020; Dabney *et al.*, 2021; Dann *et al.*, 2022]. Intrinsic rewards [Schmidhuber, 2010; Kim *et al.*, 2019; Ermolov and Sebe, 2020; Bougie and Ichise, 2020; Mu *et al.*, 2022; Henaff *et al.*, 2022; Mutti *et al.*, 2022] driven by state uncertainty or novelty measurement incentivize and guide the direction of exploration. Entropy has been empirically used in policy optimization to directly encourage exploration [Ahmed *et al.*, 2019; Haarnoja *et al.*, 2018; Zhang *et al.*, 2024; Han and Sung, 2021]. However, existing methods pay less attention to the balance between exploration and exploitation, either leaning toward one side of the exploration-exploitation spectrum or relying on manually designed heuristics to gradually reduce exploration. None of them has achieved a principled adaptation between exploration and exploitation.

In this paper, we revisit the exploration-exploitation dilemma from an entropy-driven perspective. We start by elucidating the relationship between entropy and exploratory motivation, using entropy as a proxy to understand the trade-off between exploration and exploitation. We then prove that an extended Bellman equation with a novelty-seeking term guarantees appropriate entropy adjustment and its globally monotonic decline. Hence, we theoretically formulate a dynamic adaptive process of policy optimization that accommodates all three cases of the exploration-exploitation trade-off and ultimately transits from exploration to exploitation.

Based on the theoretical findings, we instantiate an end-to-end adaptive framework *AdaZero*, which adaptively balances the strength of exploration and exploitation. A state autoencoder takes real-time state images as input and outputs the reconstruction error as an exploratory reward to encourage exploration. The reconstructed state image generated by the autoencoder, which represents the agent's mastery level of the current state, is fed into an evaluation network to determine the strength of exploration and exploitation. Guided by this mechanism, agents can automatically adapt to environmental changes and decide how much to explore or exploit.

To provide empirical support for our theoretical findings and validate the effectiveness of our proposed self-adaptive mechanism, we conducted experiments on many Atari and MuJoCo environments, considering both continuous and discrete action state spaces. The results indicate that AdaZero tends to outperform previous methods across environments ranging from simple to complex, highlighting the potential effectiveness of our self-adaptive mechanism. Especially in Montezuma, AdaZero boosts the final returns by up to fif-

*Corresponding author

teen times. We also perform a series of visualization analyses to reveal the functionality of entropy in policy optimization and how AdaZero's automatic adaptive exploration and exploitation effectively works. In summary, this paper makes the following three innovations:

- It establishes a theory basis for quantifying exploration-exploitation trade-off through the lens of entropy.
- It proves an optimization scheme naturally accommodating trade-off with global convergence of exploration, guiding the design of a principled self-adaptive policy.
- It instantiates the theoretical findings as an adaptive framework AdaZero and validates its strong generalizability through extensive experiments.

2 Related Work

2.1 Sparse Rewards Tasks

Sparse reward [Gallouédec and Dellandréa, 2023; Kim *et al.*, 2023; Silvia, 2012; Barto, 2013; Bellemare *et al.*, 2016; Ostrovski *et al.*, 2017; Pathak *et al.*, 2017; Burda *et al.*, 2018; Yan *et al.*, 2024] is a typical challenge in reinforcement learning and is widespread in real-world applications such as robot control, autonomous driving, gaming, and complex task planning. The characteristic of such environments is that the agent has limited opportunities to receive reward signals unless it reaches specific goals or triggers certain events. As a consequence, the agent receives little effective feedback most of the time, leading to a drastic reduction in policy optimization efficiency and making it difficult to determine which actions or state sequences lead to the eventual reward [Mnih *et al.*, 2013; Watkins, 1989]. Therefore, enhancing exploration in sparse-reward environments remains a critical challenge in reinforcement learning.

2.2 Exploration-Enhancing Methods

In sparse-reward environments, mainstream approaches can be broadly divided into two categories: methods based on maximum entropy and methods that use intrinsic rewards, both of which aim to expand action and/or state exploration.

The core idea of entropy-based reinforcement learning [Arriojas *et al.*, 2023; Haarnoja *et al.*, 2018; Ahmed *et al.*, 2019; Zhang *et al.*, 2024; Han and Sung, 2021] is to apply entropy regularization to policy optimization in order to maintain policy diversity. By incorporating an entropy term into the objective function, agents are encouraged to explore a broader range of actions, thereby avoiding local optima. This mechanism enhances the agent's exploration capabilities, making it more adaptable in sparse reward environments.

Another stream of research focuses on manipulating the rewards, particularly the exploratory rewards [Burda *et al.*, 2018; Ladosz *et al.*, 2022; Chen *et al.*, 2022], which are added to the original exploitative rewards from the environment. To balance the effect of the two types of rewards, a decaying scheme is introduced for the exploratory rewards [Stadie *et al.*, 2015; Badia *et al.*, 2020; Guo *et al.*, 2022]. However, these methods fail to achieve an optimal balance between exploration and exploitation, which may lead to an effective learning process [Taiga *et al.*, 2021]. Especially when the environmental rewards are sparse, the ever-existing exploratory rewards lead to collapsed total rewards and aimless exploration.

2.3 Exploration-Exploitation Decoupled RL

Decoupled RL methods [Colas *et al.*, 2018; Schäfer *et al.*, 2021; Whitney *et al.*, 2021; Liu *et al.*, 2021], which explicitly separate the exploration and the exploitation processes and train them independently. Through this approach, the agent can focus on specific tasks during different learning phases.

However, despite the theoretical advantage of decoupled RL methods, some challenges remain in practice. A major issue is that a rigid separation and periodically switching between exploration and exploitation can lead to drop in efficiency. The agent may fail to fully utilize the information gathered in the exploration phase for exploitation, or waste resources in exploration neglecting actual strategy requirements. Moreover, the coordination mechanism of exploration and exploitation is not well designed. The agent may struggle to adapt flexibly to changes in the actual environment, making the dynamic balance between exploration and exploitation more difficult. Thus, how to effectively coordinate the alternation and collaboration between exploration and exploitation in practical applications remains to be further researched.

2.4 Exploration-Exploitation Trade-off

At the policy level, attempts have been made to incorporate exploration with exploitation, the representative of which is ϵ-greedy and its variations [Dabney *et al.*, 2021; Taiga *et al.*, 2021]. While this method generally applies to a wide range of environments, it mainly relies on manually tuned hyperparameters or a specifically designed prior.

2.5 Atari and MuJoCo Benchmarks

In RL research, Atari [Towers *et al.*, 2026] and MuJoCo [Ouhssain *et al.*, 2026] are two of the most commonly used and representative benchmark environments. Atari games are typically based on high-dimensional pixel observations and involve discrete action spaces, complex state transitions, and varying degrees of reward sparsity. Thus, they are widely used to evaluate agents in visual decision-making, long-horizon exploration, and sparse-reward scenarios. In particular, challenging tasks such as Montezuma's Revenge require long sequences of effective exploration to obtain rewards, making them important platforms for examining exploration mechanisms.

In contrast, MuJoCo focuses on continuous control tasks, where agents are required to learn stable, smooth, and high-precision control policies in physical simulation environments. These tasks involve continuous state and action spaces, with reward signals usually related to motion efficiency, posture stability, energy consumption, or distance to the target. Therefore, MuJoCo is well suited for evaluating continuous-action decision-making, dynamic system modeling, and policy optimization stability. Using Atari and MuJoCo together enables a comprehensive evaluation of RL algorithms across discrete and continuous control, visual decision-making and physical control, as well as sparse and dense reward settings.

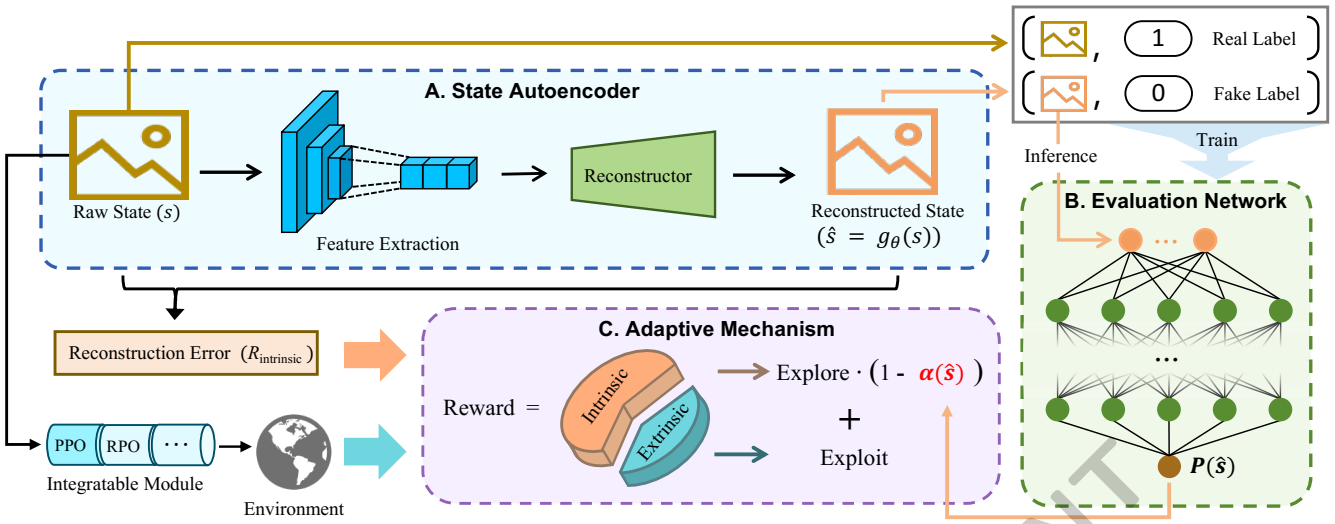


Figure 1: AdaZero's Framework. AdaZero consists of three main components: (A) State Autoencoder, (B) Evaluation Network for the level of mastery, and (C) Adaptive Mechanism. The state autoencoder encodes and reconstructs states in raw images, where the reconstruction errors drive the agent's exploration. The mastery evaluation network evaluates the reconstructed states and outputs the probability of $\hat{s}$ being real images as the balance factor $\alpha(\hat{s})$. Finally, $\alpha(\hat{s})$ is used in the adaptive mechanism to balance exploration and exploitation dynamically.

3 Preliminaries

Reinforcement Learning. Reinforcement Learning [Yan *et al.*, 2025; Wu *et al.*, 2024; Zhao *et al.*, 2021] is commonly modeled as a Markov Decision Process (MDP) [Sutton and Barto, 1999; Wu *et al.*, 2023], it is described by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ represent state and action spaces respectively. The function P is the transition probability function, indicating the likelihood of moving from one state to another following an action. The function R is the reward function. The discount factor $\gamma \in [0, 1)$ modifies the value of future rewards. The agent's interactions within the MDP generate a trajectory τ of states, actions, and rewards: $s_0, a_0, R(s_0, a_0), \cdots, s_t, a_t, R(s_t, a_t), \cdots$. The state-action value function $Q^\pi(s_t, a_t)$ is defined as follows:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{i=0}^{\infty} \gamma^i R(s_{t+i}, a_{t+i}) \mid s_t, a_t\right]. \quad (1)$$

Let $\pi(a|s)$ be the policy determining the probability of taking action a in state s. The policy is defined in the softmax form of the Q value:

$$\pi(a_i|s) = \frac{e^{Q(s,a_i)}}{\sum_k e^{Q(s,a_k)}}, i = 1, 2, \cdots, n. \quad (2)$$

The entropy of the policy π is defined as:

$$H(\pi|s) = -\sum_{a_i \in \mathcal{A}} \pi(a_i|s) \log \pi(a_i|s).$$

To guide the agent to make decisions that maximize long-term gains, the goal of policy optimization is to learn a policy π that maximizes the expected total discounted reward $\eta(\pi)$, defined as follows:

$$\eta(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{i=0}^{\infty} \gamma^i R(s_i, a_i)\right].$$

Exploration-Exploitation Dilemma. Extending the traditional MDP framework to incorporate exploration, new rewards are introduced to help agents explore sparse reward tasks. The total reward function R_{total} combines exploitative rewards R_{exploit} (extrinsic rewards corresponding to above R in MDP) and exploratory rewards R_{explore} (intrinsic rewards):

$$R_{\text{total}}(s, a) = R_{\text{exploit}}(s, a) + \lambda R_{\text{explore}}(s, a).$$

The Q-function is updated with exploratory rewards added:

$$\begin{aligned} Q_{\text{total}}(s, a) =& E_{\tau\sim\pi}[\sum_i \gamma^i (R_{\text{exploit}}(s_i, a_i) \\ &+ \lambda R_{\text{explore}}(s_i, a_i))|s, a] \\ \triangleq& Q_{\text{exploit}}(s, a) + \delta(s, a), \end{aligned} \quad (3)$$

where $Q_{\text{exploit}}(s, a) = E_\tau\left[\sum_i \gamma^i R_{\text{exploit}}(s_i, a_i)|s, a\right]$, and $\delta(s, a) = \lambda E_\tau\left[\sum_i \gamma^i R_{\text{explore}}(s_i, a_i)|s, a\right]$ is the accumulated return with only exploratory rewards for simplicity.

Note that the coefficient λ is empirically determined in existing research due to the inherent challenge of the exploration-exploitation trade-off. Moreover, as R_{exploit} is generally sparse in practice, optimization will be biased toward R_{explore}, leading to a non-optimal policy.

4 Exploration-Exploitation Adaptation from the Perspective of Entropy

Probing Exploration-Exploitation Trade-off Through Entropy. We first investigate the relationship between entropy and exploratory behavior. According to Eqn. (2), we can define the policies without and with exploratory motivation, i.e., π_{exploit} and π_{total} respectively, which are the softmax over Q_{exploit} and Q_{total}. We prove that under mild conditions, entropy and exploratory rewards consistently increase or decrease together during policy optimization. Given an action

space $\mathcal{A}$ of N actions, suppose $a_1 \in \mathcal{A}$ is the optimal action and $a_2 \in \mathcal{A}$ is the suboptimal action, and so on, i.e. $Q_{\text{exploit}}(s, a_1) \geq Q_{\text{exploit}}(s, a_2) \geq \cdots \geq Q_{\text{exploit}}(s, a_N)$. For the convenience of subsequent proofs, we define the following abbreviations for the commonly used terms:

$$\begin{aligned} &H(\pi_{\text{total}}, \delta_1, \delta_2, \cdots, \delta_N|s) \triangleq H(\pi_{\text{total}}|s) \\ =&-\sum_{i=1}^{N} \frac{e^{Q_i+\delta_i}}{\sum_j e^{Q_j+\delta_j}} \log \frac{e^{Q_i+\delta_i}}{\sum_j e^{Q_j+\delta_j}} \\ =&-\sum_{i=1}^{N} \frac{e^{Q_i+\delta_i}}{\sum_j e^{Q_j+\delta_j}} (Q_i + \delta_i) + \log \sum_j e^{Q_j+\delta_j}, \end{aligned}$$

where $Q_i = Q_{\text{exploit}}(s, a_i)$ and $\delta_i = \delta(s, a_i)$.

Lemma 4.1. *For $\forall j > i$, if $0 \leq \delta(s, a_j) - \delta(s, a_i) \leq Q_{exploit}(s, a_i) - Q_{exploit}(s, a_j)$ holds, for any s, we have*

$$H(\pi_{total}, \delta_1, \delta_2, \cdots, \delta_{i-1}, \overbrace{\delta_{i-1}, \cdots, \delta_{i-1}}^{N-i+1}|s) \leq$$
$$H(\pi_{total}, \delta_1, \delta_2, \cdots, \delta_{i-1}, \delta_i, \overbrace{\delta_i, \cdots, \delta_i}^{N-i}|s).$$

Lemma 4.2. *(Connection between entropy and exploration.) Under the condition of Lemma 4.1, for any s, we have*

$$H(\pi_{exploit}|s) \leq H(\pi_{total}|s).$$

The proofs of Lemma 4.1 and 4.2 are shown in Appendix. Lemma 4.2 demonstrates that incorporating exploratory rewards leads to increased entropy, thereby enhancing the agent's exploration capability. Hence, we gain a theoretical understanding of the exploration-exploitation trade-off through the lens of entropy.

The condition of Lemma 4.1 indicates that when $\delta(a_j) - \delta(a_i) > 0$, the probability of non-optimal actions increases, thereby encouraging the exploration of more non-optimal actions. However, the exploration should not be aimless. Thus, we assume its upper bound to be $Q_{exploit}(a_i) - Q_{exploit}(a_j)$. The above condition is mild and has already been satisfied in existing classic methods like discrete SAC [Haarnoja *et al.*, 2018] and RND [Burda *et al.*, 2018].

Self-Adaptive Bellman Equation. As depicted in Eqn. 3, in current RL methods with a fixed coefficient, the exploration continues to play a role even when it is no longer necessary. Hence, we formulate a new adaptive mechanism for dynamically balancing exploration and exploitation. We theoretically prove how its exploration can be adjusted towards both sides of the exploration-exploitation spectrum in a principled manner.

Specifically, we extend the original Bellman equation with a novelty-seeking term:

$$\begin{aligned} Q_{\text{total}}(s, a) =& E_\tau[\sum_i \gamma^i (R_{\text{exploit}}(s_i, a_i) \\ &+ (1-\alpha(s_i)) R_{\text{explore}}(s_i, a_i))|s, a] \\ \triangleq& Q_{\text{exploit}}(s, a) + \hat{\delta}(s, a), \end{aligned} \tag{4}$$

where $Q_{\text{exploit}}(s, a)$ is the same as in Eqn. (3), $\hat{\delta}(s, a) = E_\tau\left[\sum_i \gamma^i (1 - \alpha(s_i)) R_{\text{explore}}(s_i, a_i)|s, a\right]$ represents the strength of exploration, and $\alpha(s_i) \in [0, 1]$ is a function to balance the trade-off between exploration and exploitation on each state s_i.

Based on Eqn. (4), we can derive a policy optimization process that inherently accommodates all three regimes of the exploration-exploitation spectrum from the perspective of entropy. In contrast to our method, as shown in Lemma 4.2, existing methods can only maximize entropy to foster exploration without the capability to adaptively increase or decrease the strength of exploration.

Theorem 4.1. *Under conditions of Lemma 4.1 and different values of α function, for any s, policy optimization with the above Eqn. 4 can have either*

$$H(\pi_{exploit}|s) \leq H(\pi_{total}|s) \text{ or } H(\pi_{exploit}|s) > H(\pi_{total}|s).$$

Proof. According to Eqn. (4), we have

$$\hat{\delta}(s, a) = (1-\alpha(s)) R_{\text{explore}}(s, a) + \gamma \mathbb{E}_{s', a'|s, a}[\hat{\delta}(s', a')]. \tag{5}$$

Depending on the mastery level of the states $\alpha(s)$, there will be different scenarios regarding the relationship between exploration and exploitation. We mainly discuss the following three typical cases:

Case I - Exploration Dominant: for any s, when $\alpha(s) \equiv 0$, $Q_{\text{total}}(s, a_1) = Q_{\text{exploit}}(s, a_1) + \hat{\delta}(s, a_1)$, and $Q_{\text{total}}(s, a_2) = Q_{\text{exploit}}(s, a_2) + \hat{\delta}(s, a_2)$. From lemma 4.1, we know $H(\pi_{\text{exploit}}|s) < H(\pi_{\text{total}}|s)$, i.e., the entropy increases. In this scenario, the strength of exploration reaches the peak level, which means our method exhibits stronger exploratory capabilities than methods without intrinsic rewards.

Case II - Adaptive Exploration-Exploitation: According to Eqn. (5), let $\alpha(s) = 1$, we have

$$\begin{aligned} Q_{total}(s, a_1) =& Q_{exploit}(s, a_1) + \hat{\delta}(s, a_1) \\ =& Q_{exploit}(s, a_1) + \gamma E_{s', a'|s, a_1}[\hat{\delta}(s', a')], \end{aligned}$$

$$\begin{aligned} Q_{total}(s, a_2) =& Q_{exploit}(s, a_2) + \hat{\delta}(s, a_2) \\ =& Q_{exploit}(s, a_2) + \gamma E_{s', a'|s, a_2}[\hat{\delta}(s', a')]. \end{aligned}$$

Here we discuss two cases. For the first case, when the subsequent transition state space corresponding to action a_1 differs from that of action a_2, the following holds:

$$E_{s', a'|s, a_2}[\hat{\delta}(s', a')] = 0$$

and

$$E_{s', a'|s, a_1}[\hat{\delta}(s', a')] \neq 0.$$

For the second case, If the subsequent transition state spaces corresponding to a_1 and a_2 are the same, except that the probabilities for each subsequent transition state differ. We can assume two distinct states s' and s'' where the probabilities diverge. Let $p(s')$, $p(s'')$ and $\hat{p}(s')$, $\hat{p}(s'')$ denote the transition probabilities for a_1 and a_2, respectively, while all other probabilities (denoted collectively as C remain identical). Assume $p(s') < \hat{p}(s')$ and $p(s'') > \hat{p}(s'')$. We then have

$$\begin{aligned} Q_{\text{total}}(s, a_1) =& Q_{\text{exploit}}(s, a_1) + \gamma (p(s') \hat{\delta}(s') (1 - \alpha(s')) \\ &+ p(s'') \hat{\delta}(s'') (1 - \alpha(s'')) + C) \end{aligned}$$

$$Q_{\text{total}}(s,a_2) = Q_{\text{exploit}}(s,a_2) + \gamma(\hat{p}(s')\hat{\delta}(s')(1-\alpha(s')) + \hat{p}(s'')\hat{\delta}(s'')(1-\alpha(s'')) + C).$$

By Lemma 4.1, to prove that the entropy decreases, it suffices to show the existence of $\alpha(s')$ and $\alpha(s'')$ such that

$$p(s')\hat{\delta}(s')(1-\alpha(s')) + p(s'')\hat{\delta}(s'')(1-\alpha(s'')) > \hat{p}(s')\hat{\delta}(s')(1-\alpha(s')) + \hat{p}(s'')\hat{\delta}(s'')(1-\alpha(s'')).$$

To prove this inequality, we reformulate it as

$$(p(s'')-\hat{p}(s''))\hat{\delta}(s'')(1-\alpha(s'')) > (\hat{p}(s')-p(s'))\hat{\delta}(s')(1-\alpha(s')).$$

Simplifying further yields

$$1-\alpha(s') < \frac{(p(s'')-\hat{p}(s''))\hat{\delta}(s'')}{(\hat{p}(s')-p(s'))\hat{\delta}(s')}(1-\alpha(s'')).$$

Apparently, there exist $\alpha(s')$ and $\alpha(s'')$ that satisfies $Q_{\text{total}}(s,a_1) = Q_{\text{exploit}}(s,a_1) + \hat{\delta}(s,a_1)$ and $Q_{\text{total}}(s,a_2) = Q_{\text{exploit}}(s,a_2)$. This means $H(\pi_{\text{exploit}}|s) > H(\pi_{\text{total}}|s)$, i.e., the probability of optimal policy increases while entropy starts to decrease. In this scenario, the degree of exploitation adaptively increases while the level of exploration decreases.

Case III - Exploitation Dominant: for any s, when $\alpha(s) \equiv 1$, we have $Q_{\text{total}}(s,a_1) = Q_{exploit}(s,a_1)$, and $Q_{\text{total}}(s,a_2) = Q_{\text{exploit}}(s,a_2)$. This means the entropy remains unchanged, i.e., $H(\pi_{\text{exploit}}|s) = H(\pi_{\text{total}}|s)$. In this scenario, the strength of exploration is zeroed out and exploitation is dominant. □

From the above theorem, we know that as $\alpha(s)$ changes, the entropy of our policy can get higher or lower than that of π_{exploit}. This implies that our method can conduct a dynamic level of exploration and adaptively alternate between exploration and exploitation. Visual evidence is provided in Fig. 3(b). Our design also aligns with the principle of Occam's Razor [Murphy, 2012; Domingos, 2012] in that exploration rewards should not be posited without necessity.

Guarantee of Entropy Minimization. We further prove that based on Eqn. (4), the entropy of policy is guaranteed to decrease in the global perspective.

Theorem 4.2. *(Decay of entropy after optimization.) During the optimization, the sequences $\{\hat{\delta}_{k-1}(s,a_i)\}_{i=1}^N$ at step k and the sequences $\{\hat{\delta}_k(s,a_i)\}_{i=1}^N$ at step $k+1$ satisfy the conditions of Lemma 4.1, in addition to the following relation, for $i \le j$:*

$$0 \le \hat{\delta}_{k-1}(s,a_i) - \hat{\delta}_k(s,a_i) \le \hat{\delta}_{k-1}(s,a_j) - \hat{\delta}_k(s,a_j),$$

we have

$$H(\pi_{total}, \hat{\delta}_1^k, \cdots, \hat{\delta}_N^k|s) \le H(\pi_{total}, \hat{\delta}_1^{k-1}, \cdots, \hat{\delta}_N^{k-1}|s).$$

The proof of Theorem 4.2 can be obtained by applying Lemma 4.1 and is shown in Appendix. The steady decrease of entropy is also validated through visualization of entropy curves in Fig. 3. Note that Theorem 4.2 applies to a broad spectrum of RL methods with intrinsic motivation that approaches zero during training.

5 Validation of Adaptive Theory: AdaZero

Inspired by the above theoretical findings, we formalize an exploration-exploitation dynamic adaptive framework named AdaZero. As illustrated in Fig. 1, AdaZero consists of three parts: a state autoencoder, a mastery evaluation network, and an adaptive policy optimization mechanism.

State Autoencoder. The state autoencoder aims to motivate the agent's exploratory behavior of searching for unknown policies to get familiar with the environment. As shown in Fig. 1A, the state autoencoder receives raw states s via the interaction between the agent and the environment and learns to reconstruct the input images $\hat{s} = g_\theta(s)$, where θ is the network parameters. The reconstruction error $R_{\text{explore}}(s,a)$ is defined as follows:

$$R_{\text{explore}}(s,a) = \mathcal{L}_g(\theta) = \frac{1}{2}\left\|s - g_\theta(s)\right\|_2^2.$$

We use the reconstruction error to drive exploration. A larger error indicates insufficient network training on the state, and hence, exploration needs to be enhanced for this state. Conversely, a smaller error indicates that the training on that state has been sufficient, leading to an automatic decrease in exploratory rewards.

Mastery Evaluation Network. The above state autoencoder provides information about where to explore. Furthermore, determining whether or not and how much to explore based on the agent's mastery level of the states is also crucial. To this end, AdaZero involves a mastery evaluation network to provide real-time assessment of the necessity of exploration, as shown in Fig. 1B.

The mastery network is implemented as a binary classifier denoted as f_ω. At the training stage, the network receives mixed data of real and fake state images reconstructed by the state autoencoder, denoted as s and $\hat{s}$, respectively. It is trained to distinguish between the two kinds of inputs and output the probability of the inputs being real images, with a cross-entropy loss defined as follows:

$$\mathcal{L}_f(\omega) = -\mathbb{E}_{s\sim\mathcal{S}}[\log(f_\omega(s))] - \mathbb{E}_{\hat{s}\sim\hat{\mathcal{S}}}\left[\log\left(1-f_\omega(\hat{s})\right)\right],$$

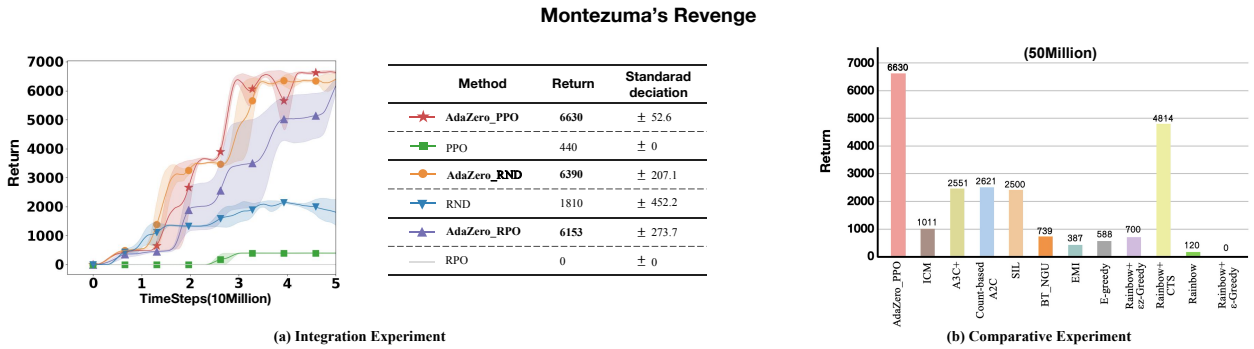
where ω is the network's parameters, $\mathcal{S}$ and $\hat{\mathcal{S}}$ are the set of real state images and reconstructed images, respectively.

At the inference stage, the evaluation network only receives the reconstructed images from the state autoencoder and outputs the probability $\alpha(\hat{s}) \triangleq f_\omega(\hat{s})$. Hence, $\alpha(\hat{s}) \in [0,1]$ can naturally serve as a measure of AdaZero's grasp of the current training state information.

Adaptive Mechanisms. Inspired by Eqn. 4 and Theorem 4.1, by integrating the reconstruction error R_{explore} provided by the state autoencoder and the estimated level of mastery of the state information $\alpha(\hat{s})$ from the evaluation network, we implement the adaptive mechanism of AdaZero depicted in Fig. 1C through the following formula:

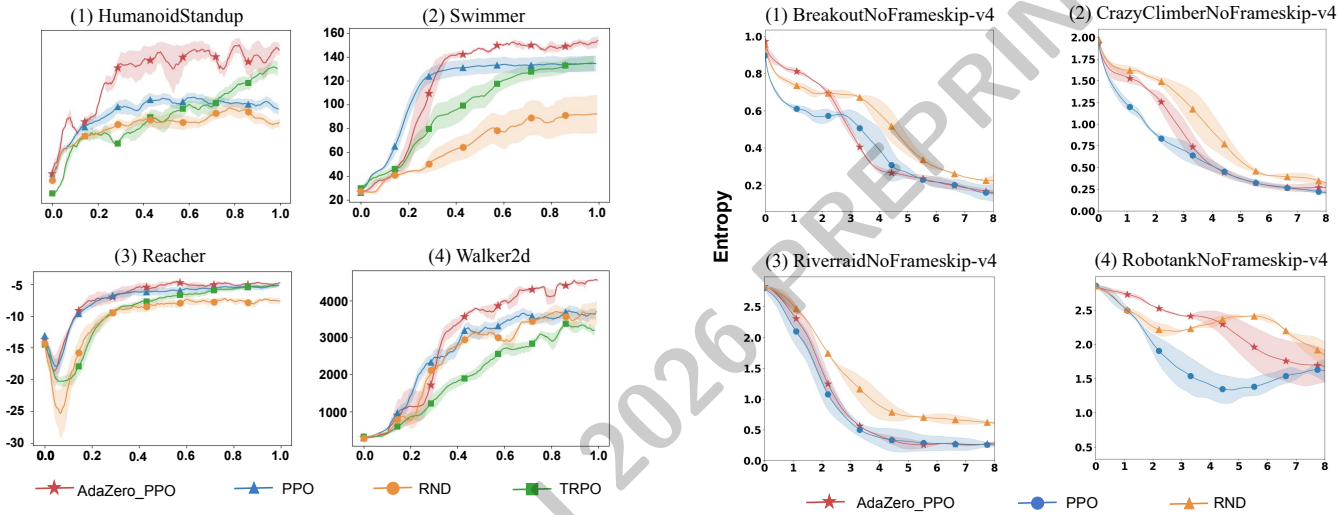
$$R_{\text{total}}(s,a) = R_{\text{exploit}}(s,a) + (1-\alpha(\hat{s}))R_{\text{explore}}(s,a). \quad (6)$$

When the agent reaches a high level of mastery in the current state, the exploration intensity should be reduced to increase the exploitation ratio. Conversely, when the level of mastery is



(a) Integration Experiment (b) Comparative Experiment

Figure 2: Main experiments in the most challenging Montezuma's Revenge. (a) We integrate AdaZero into three representative RL methods and show that AdaZero can bring improvements; (b) Our method also presents an advantageous performance compared with other advanced baselines. Equipped with AdaZero, RND reached the score published in the original paper within only one-tenth of the training steps.



(a) Rewards in MuJoCo. (b) Entropy Visualization.

Figure 3: (a) The results show the generalizable performance of AdaZero in different types of continuous-space tasks. In this figure, the red curve denotes AdaZero_PPO, the blue curve denotes PPO, the orange curve denotes RND, and the green curve denotes TRPO. (b) Entropy Visualization of AdaZero vs. PPO and RND in Atari. The x-axis represents timesteps in millions.

low, the exploration intensity should be increased to encourage the agent to better learn from the various information about the state. As $\alpha(\hat{s}) \in [0, 1]$ reflects the agent's ability to grasp valuable information in the reconstructed states, we use $(1 - \alpha(\hat{s}))$ to control the strength of intrinsic rewards adaptively, hence dynamically balance the exploration and exploitation.

Compared with methods that decay intrinsic rewards [Badia *et al.*, 2020], the adaptive mechanism of AdaZero is fulfilled by end-to-end networks. According to Theorem 4.1, from a local perspective, the exploratory incentive of AdaZero can be adaptively increased or decreased, or completely eliminated in response to the training progress and environment changes. Furthermore, according to Theorem 4.2, the entropy of AdaZero decreases globally, thereby theoretically guaranteeing the eventual transition from exploration to exploitation.

6 Experiments

6.1 Main Experiments

To provide empirical support for our theoretical findings and validate the effectiveness of AdaZero framework, we conduct experiments on a variety of RL tasks, ranging from dense to sparse reward environments, continuous to discrete action spaces, and quantitative comparisons to visual analyses.

We compare AdaZero with the recent competitive and widely-used RL algorithms, including PPO [Schulman *et al.*, 2017], RND [Burda *et al.*, 2018], and RPO [Gan *et al.*, 2024]. For continuous tasks, we additionally include TRPO [Schulman *et al.*, 2015] as our baseline. As AdaZero is designed to be flexible and can be easily integrated into existing algorithms, we first compare the performance of the baselines integrated with AdaZero versus without AdaZero. Then, we focus on PPO with AdaZero as a representative of our method.

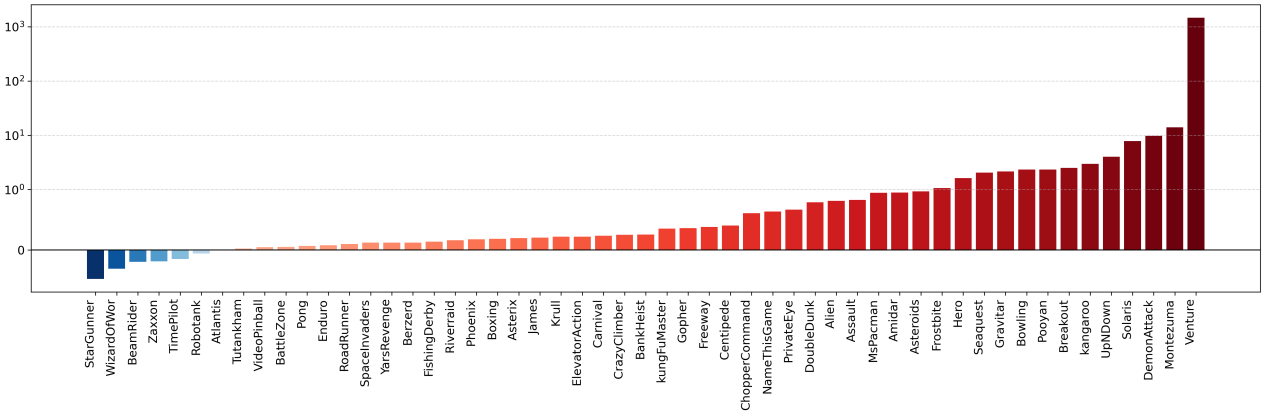


Figure 4: Summary of the results of AdaZero compared with PPO across Atari environments. The y-axis denotes the normalized improvement of AdaZero against the baseline, following [Gan *et al.*, 2024]. This experiment uses more random seeds than the other experiments and reports the averaged performance across these seeds.

Table 1: Comparison of **reward acquisition efficiency**. We compare the time taken by our method and the RND method to achieve the same reward (Threshold). Our method requires significantly less training time than RND to reach the same reward.

Metrics \ Environments	Montezuma's Revenge	Hero	Frostbite	Gravitar	Alien
Threshold	2000	10000	2000	500	1500
Efficiency gains	27%	69%	35%	74%	14%

In the remainder of this paper, AdaZero refers to PPO integrated with AdaZero. Note that RND integrated with AdaZero (AdaZero_RND) is implemented by using RND's intrinsic reward module and adapting its magnitude through the weights predicted by AdaZero's autoencoder and evaluation network.

We conducted integration experiments in the most challenging RL environment, Montezuma's Revenge [Bellemare *et al.*, 2016]. The results are shown in Fig. 2. Overall, AdaZero boosts the total returns up to about fifteen times for PPO and 3.5 times for RND in Fig. 2 (a). Besides, AdaZero significantly outperforms other baselines in Fig. 2 (b). It demonstrates that within the same training steps, AdaZero surpasses other algorithms in final returns.

The integration experiment results in Fig. 2 (a) also confirm AdaZero's effectiveness in collaboration with different methods and in highly challenging environments. Compared with algorithms like PPO and RPO, which primarily focus on exploitation, AdaZero maintains a high exploitation rate and supports selective exploration. Compared with the exploration-centric RND, AdaZero enhances its exploitation ability. Moreover, it can be observed that there are more horizontal fluctuations in the three curves corresponding to AdaZero. This indicates that our dynamic adaptive mechanism enables the agent to discover more localized policies during training, thereby enhancing the overall training efficiency. Without implementing the proposed AdaZero, PPO and RPO would be trapped in invalid policies and almost fail.

6.2 Generalization Experiments

To demonstrate AdaZero's robustness and broad applicability, this subsection presents its performance on extensive environments [Taiga *et al.*, 2020], including other Atari games as well as continuous-space environments of MuJoCo. Fig. 4 indicates that AdaZero is robust in challenging discrete-space tasks. Fig. 3 (a) shows the performance of AdaZero in continuous-action tasks. The results indicate that our algorithm outperforms baseline algorithms in MoJuCo environments, confirming that AdaZero is broadly effective in continuous spaces.

6.3 Computational Cost and Efficiency

Our exploration network size is similar to that of RND. RND requires two encoders of equal size (2M x 2 parameters) for exploration reward evaluation, while our exploration network ranges from 2 to 4M (size varies with feature layer). The evaluation network is a convolutional network with 0.5M parameters. We recorded the total training time across all environments. Compared to RND, our method requires about 1.3 times the time to reach the same total steps, mainly due to the evaluation network's involvement in training. However, our method achieves the same score in less time, showing higher time efficiency. In Atari, we recorded an average improvement of about **43%** in time efficiency at the same score. Time efficiency is calculated as: Time efficiency $= \frac{t_{\text{RND}} - t_{\text{AdaZero}}}{t_{\text{RND}}}$. The results are shown in **Tab. 1**.

7 Conclusion

This paper reveals the role of entropy in policy optimization. Based on this motivation, we re-examined the exploration-exploitation dilemma from an entropy perspective and proposed an adaptive exploration-exploitation mechanism. Inspired by this mechanism, we formalized AdaZero, which adapts using end-to-end networks. Both theoretical analysis and experimental results demonstrate that our method enables selective exploration while maintaining effective exploitation, outperforming traditional RL algorithms across diverse environments. In future work, we will extend the adaptive theory beyond decision-making tasks.

Acknowledgements

We would like to thank all the reviewers for their constructive comments. This work was supported by the NSFC (62341407, U24B20166, 62495102), Beijing Natural Science Foundation (Grant L223004), and "111" Project (B18001).

References

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

Argenis Arriojas, Jacob Adamczyk, Stas Tiomkin, and Rahul V Kulkarni. Entropy regularized reinforcement learning using large deviation theory. *Physical Review Research*, 2023.

Adria Puigdomenech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martin Arjovsky, Alexander Pritzel, Andrew Bolt, et al. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, pages 1–28, 2020.

Andrew G Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pages 17–47. 2013.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Nicolas Bougie and Ryutaro Ichise. Towards high-level intrinsic exploration in reinforcement learning. In *International Joint Conference on Artificial Intelligence*, pages 5186–5187, 2020.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Eric Chen, Zhang-Wei Hong, Joni Pajarinen, and Pulkit Agrawal. Redeeming intrinsic rewards via constrained optimization. *Advances in Neural Information Processing Systems*, 35:4996–5008, 2022.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. Geppg: Decoupling exploration and exploitation in deep reinforcement learning algorithms. In *International conference on machine learning*, pages 1039–1048. PMLR, 2018.

Will Dabney, Georg Ostrovski, and Andre Barreto. Temporally-extended ε-greedy exploration. In *International Conference on Learning Representations*, 2021.

Chris Dann, Yishay Mansour, Mehryar Mohri, Ayush Sekhari, and Karthik Sridharan. Guarantees for epsilon-greedy reinforcement learning with function approximation. In *International conference on machine learning*, pages 4666–4689. PMLR, 2022.

Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

Aleksandr Ermolov and Nicu Sebe. Latent world models for intrinsically motivated exploration. In *Advances in Neural Information Processing Systems*, pages 5565–5575, 2020.

Quentin Gallouédec and Emmanuel Dellandréa. Cell-free latent go-explore. In *International Conference on Machine Learning*, pages 10571–10586, 2023.

Yaozhong Gan, Renye Yan, Zhe Wu, and Junliang Xing. Reflective policy optimization. *International Conference on Machine Learning*, 2024.

Zhaohan Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, et al. Byol-explore: Exploration by bootstrapped prediction. *Advances in neural information processing systems*, 35:31855–31870, 2022.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Seungyul Han and Youngchul Sung. A max-min entropy framework for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:25732–25745, 2021.

Mikael Henaff, Roberta Raileanu, Minqi Jiang, and Tim Rocktäschel. Exploration via elliptical episodic bonuses. In *Advances in Neural Information Processing Systems*, pages 1–10, 2022.

Hyoungseok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. EMI: Exploration with mutual information. In *International Conference on Machine Learning*, pages 3360–3369, 2019.

Woojun Kim, Jeonghye Kim, and Youngchul Sung. LESSON: Learning to integrate exploration strategies for reinforcement learning via an option framework. *arXiv preprint arXiv:2310.03342*, 2023.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

Evan Z Liu, Aditi Raghunathan, Percy Liang, and Chelsea Finn. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *International conference on machine learning*, pages 6925–6935. PMLR, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah Goodman, Tim Rocktäschel, and Edward Grefenstette. Improving intrinsic exploration with language abstractions. In

Advances in Neural Information Processing Systems, pages 1–14, 2022.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*, pages 16223–16239, 2022.

Georg Ostrovski, Marc G Bellemare, Aaron Van Den Oord, and Remi Munos. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pages 2721–2730, 2017.

Asmae Ouhssain, Sébastien Harispe, Mohamed-Harith Ibrahim, Joseph Diab, Denis Mottet, and Jacky Montmain. Reinforcement learning with musculoskeletal models to study fatigue effects on human muscle synergies. In *Proceedings of the 10th International Conference on Movement and Computing*, pages 1–9, 2026.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787, 2017.

Lukas Schäfer, Filippos Christianos, Josiah Hanna, and Stefano V Albrecht. Decoupling exploration and exploitation in reinforcement learning. In *ICML 2021 Workshop on Unsupervised Reinforcement Learning*, 2021.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Paul J Silvia. Curiosity and motivation. *The Oxford handbook of human motivation*, pages 157–166, 2012.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. *Robotica*, 17(2):229–235, 1999.

Adrien Ali Taiga, William Fedus, Marlos C. Machado, Aaron Courville, and Marc G. Bellemare. On bonus based exploration methods in the arcade learning environment. In *International Conference on Learning Representations*, 2020.

Adrien Ali Taiga, William Fedus, Marlos C Machado, Aaron Courville, and Marc G Bellemare. On bonus-based exploration methods in the arcade learning environment. *arXiv preprint arXiv:2109.11052*, 2021.

Michel Tokic. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual conference on artificial intelligence*, pages 203–210. Springer, 2010.

Mark Towers, Ariel Kwiatkowski, John Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun Kg, Rodrigo Perez-Vicente, et al. Gymnasium: A standard interface for reinforcement learning environments. *Advances in Neural Information Processing Systems*, 38, 2026.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

William F Whitney, Michael Bloesch, Jost Tobias Springenberg, Abbas Abdolmaleki, Kyunghyun Cho, and Martin Riedmiller. Decoupled exploration and exploitation policies for sample-efficient reinforcement learning. *arXiv preprint arXiv:2101.09458*, 2021.

Zhe Wu, Haofei Lu, Junliang Xing, You Wu, Renye Yan, Yaozhong Gan, and Yuanchun Shi. Pae: Reinforcement learning from external knowledge for efficient exploration. In *The Twelfth International Conference on Learning Representations*, 2023.

You Wu, Zhe Wu, Renye Yan, Pin Tao, and Junliang Xing. Absorb what you need: Accelerating exploration via valuable knowledge extraction. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2024.

Renye Yan, Yaozhong Gan, You Wu, Junliang Xing, Ling Liang, Yeshang Zhu, and Yimao Cai. Adamemento: Adaptive memory-assisted policy optimization for reinforcement learning. *arXiv preprint arXiv:2410.04498*, 2024.

Renye Yan, Jikang Cheng, Yaozhong Gan, Shikun Sun, You Wu, Yunfan Yang, Liang Ling, Jinlong Lin, Yeshuang Zhu, Jie Zhou, et al. Entropy-adaptive diffusion policy optimization with dynamic step alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1924–1934, 2025.

Renye Yan, Jikang Cheng, Shikun Sun, Yi Sun, You Wu, Wei Peng, Zongwei Wang, Ling Liang, Junliang Xing, and Yimao Cai. Do less, achieve more: Do we need every-step optimization for rl fine-tuning of diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16561–16571, 2026.

Ruoqi Zhang, Ziwei Luo, Jens Sjölund, Thomas B Schön, and Per Mattsson. Entropy-regularized diffusion policy with q-ensembles for offline reinforcement learning. *arXiv preprint arXiv:2402.04080*, 2024.

Enmin Zhao, Renye Yan, Kai Li, Lijuan Li, and Junliang Xing. Learning to play hard exploration games using graph-guided self-navigation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2021.

Enmin Zhao, Renye Yan, Jinqiu Li, Kai Li, and Junliang Xing. Alphaholdem: High-performance artificial intelligence for heads-up no-limit poker via end-to-end reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4689–4697, 2022.