IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 December 2023, accepted 21 December 2023, date of publication 28 December 2023,
date of current version 5 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347803

RESEARCH ARTICLE

Boosting Policy Learning in Reinforcement Learning via Adaptive Intrinsic Reward Regulation

QIAN ZHAO, (Member, IEEE), JINHUI HAN, AND MAO XU
School of Physics and Telecommunication Engineering, Zhoukou Normal University, Zhoukou 466001, China

Corresponding author: Qian Zhao (qianzh@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 62003381 and Grant U1813202, in part by the National Key Research and Development Program of China under Grant 2020YFB1313600, and in part by the Foundation of the Science and Technology Department of Henan Province under Grant 212102210525 and Grant 222102210077.

ABSTRACT Reinforcement learning employs heuristic intrinsic rewards to facilitate effective learning and exploration of the environment by intelligent agents. Particularly in environments with sparse rewards, it's challenging for agents to reach goals or obtain extrinsic rewards through random exploration. Appropriate intrinsic rewards can significantly boost learning. However, using intrinsic rewards requires a careful balance between exploration and exploitation, which is typically adjusted by a coefficient. In addition, different settings of the intrinsic reward coefficient can lead to significant differences in learning efficiency and performance. Hence, this paper presents a novel approach to regulate intrinsic rewards by adaptively tuning their coefficients, with the aim of enhancing the performance of some existing intrinsic reward techniques. The primary contributions of this study can be summarized in three aspects: 1) Designing a coefficient that adjusts the magnitude of intrinsic rewards, which dynamically adapts based on the return curve. 2) Developing an episode-wise adjustment strategy to improve the sample efficiency of intrinsic reward methods. 3) Modifying the advantage function in gradient policy methods to mitigate training instability caused by changes in the regulated intrinsic rewards. To evaluate the proposed method, we conducted experiments in both 2D and 3D maze environments with sparse rewards, combining it with several intrinsic reward approaches. The results demonstrate that the proposed method effectively enhances learning efficiency and improves the performance of some existing approaches to a certain extent.

INDEX TERMS Exploration, intrinsic reward, reinforcement learning, learning efficiency.

I. INTRODUCTION

Reinforcement learning (RL) involves an agent learning to optimize its actions by maximizing cumulative rewards through a trial-and-error strategy [1]. However, in environments with sparse rewards, the agent struggles to acquire immediate rewards, leading to suboptimal learning [2]. Exploration in these environments rarely reaches goal states or provides meaningful feedback. This inefficiency can result in increased interactions with the environment and even complete failure in learning. To address this challenge, researchers have heuristically proposed various intrinsic reward methods that leverage observations to generate immediate rewards, encouraging the agent's exploration of the state space and improving learning efficiency [3], [4], [5], [6], [7], [8].

To further enhance the performance of intrinsic rewards, certain challenging issues need to be addressed. The parameters of intrinsic rewards significantly influence the exploration-exploitation trade-off [1], which determines whether the agent should explore unexplored states or optimize policies based on existing data. Inappropriate parameters can cause intrinsic rewards to interfere with the effects of extrinsic rewards, resulting in inefficient learning

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Nappi.

or degraded performance. Despite some intrinsic reward schemes considering this trade-off [9], [10], many methods still rely on preset parameters.

Another concern is the potential for further optimizing the performance of heuristic intrinsic reward methods. Some researchers have explored the combination of different reward types to enhance the impact of intrinsic rewards, such as integrating episodic rewards with global rewards [9] or combining count-based rewards with other types of rewards [11], [12], [13]. However, it remains unclear whether a more universal method could enhance the efficacy of intrinsic rewards across diverse task environments.

In response to these challenges, this paper proposes a method to adjust the parameters of intrinsic rewards based on feedback from extrinsic rewards for some existing reward schemes, such as AGAC [12], COUNT [7], and RIDE [13]. Our method tackles the exploration-exploitation trade-off by regulating intrinsic rewards, utilizing variations in the extrinsic reward during the learning process as empirical evidence to guide the adjustment of intrinsic rewards.

Furthermore, we propose the use of weighted intrinsic rewards within an episode to enhance exploration in trajectory-wise reward environments, where agents receive rewards only at the end of an episode. These weights regulate the intrinsic rewards for each step in an episode, following an increasing weight series from the start of the episode. This approach acknowledges that states closer to the episode's start, which are near the starting point and therefore easily observed or frequently visited based on the agent's policy, require smaller intrinsic rewards. Conversely, states closer to the episode's end are less likely to be observed, necessitating larger intrinsic rewards. For tasks with variable episode length, this intra-episode reward adjustment mechanism enables adaptive intrinsic reward adjustments.

However, adjusting intrinsic rewards may introduce training instability. To address this issue, we enhance the policy gradient method with a weighted advantage function, to avoid disruptions to policy learning caused by the aforementioned adjustments. Our modification makes the loss function time-dependent, resembling an episodic Markov decision process [14]. Although our policy function uses states as inputs and incorporates the time steps of an episode in the policy gradient, it differs in that our method is only employed in the loss function.

In summary, the proposed method, named **Ada**ptive Intrinsic Reward **Reg**ulation (AdaReg), can be viewed as a framework that enhances the performance of existing intrinsic rewards by means of the exploration-exploitation trade-off. This trade-off encompasses two dimensions: one spanning the training process and one occurring within each episode. The regulation of intrinsic rewards is dynamically adjusted according to the current stage of training. We validate our design across various trajectory-wise reward environments with discrete action spaces and with several types of intrinsic rewards. The main contributions of this study are as follows:

1) Formulation of adaptive intrinsic reward regulation, which contains an extrinsic regulation coefficient for adjusting the magnitude of intrinsic rewards, and an episodic regulation coefficient for intra-episode adjustment that further improves the sample efficiency of intrinsic reward methods.

2) Modification of the advantage function in gradient policy methods to mitigate training instability caused by shifts in regulated intrinsic rewards.

3) Validation of the proposed method through experiments conducted on MiniGrid, a sparse-reward, procedurally generated environment, on VizDoom, a singleton 3D environment, and MiniWorld, a procedurally generated 3D maze environment. The effectiveness of each component is confirmed through ablation experiments.

The remainder of this paper is organized as follows: In Section II, we present related work on the exploration of RL agents. The relevant background is provided in Section III. Then, in Section IV, our method is described in detail. Section V presents and discusses the experimental results, demonstrating the advantages of the proposed method in procedurally generated and high-dimensional observation environments. Finally, the conclusion, limitations, and further work are presented in Section VI.

II. RELATED WORK

Exploration, a fundamental component of reinforcement learning, significantly influences sample efficiency and agent performance. In recent years, numerous effective exploration methods have been proposed and extensively researched [7], [8], [12], [13], [15].

Intrinsic rewards, inspired by the concept of intrinsic motivation in psychology, have gained popularity as a method for implementing exploration [16]. These rewards provide bonuses to an agent when it accesses novel states or augments its knowledge of the environment. Count-based methods and curiosity-based methods are two main types of intrinsic reward methods that have proven to be effective in various challenging exploration environments [4], [6], [7], [8]. The focus of this work is to enhance the learning efficiency of agents through adaptive adjustments to some existing intrinsic rewards.

Count-based methods allocate bonuses based on the number of times a state has been visited [7], [8], [11], [17], [18]. This approach is initially introduced in tabular settings by counting each state [17]. Before this, the count-based approach is already in use in some learning algorithms [19]. It is further refined with the use of pseudo-counts of states [7] and a neural network-based state density estimator [8], [18]. Episodic state count is demonstrated outstanding performance in discrete state spaces [11], and we use it as one of the baseline methods in our experiments.

Curiosity-based methods, on the other hand, assign bonuses based on the uncertainty or prediction error of environment dynamics [3], [6], [13], [20]. The Intelligent Adaptive Curiosity (IAC) method utilizes a forward dynamics

prediction model to generate intrinsic rewards [3], while the Intrinsic Curiosity Module (ICM) learns a latent space with a self-supervised inverse dynamics model and uses the prediction error of the next state in the latent space as an intrinsic reward [6]. To boost the performance in procedurally generated environments, Rewarding Impact-Driven Exploration (RIDE) uses the difference of latent representations between two consecutive states as an exploration bonus [13]. We use RIDE as a representative curiosity-based reward method in our experiments to validate the effectiveness of our approach.

Adversarial learning techniques from supervised learning have also been applied to reinforcement learning for enhancing exploration [21], [22], [23]. The Adversarially Guided Actor-Critic (AGAC) introduces an adversarial policy to learn the behavior of the current policy, rewarding actions that deviate from the adversarial policy [12]. We also consider AGAC as one of the baseline methods in our experiments for comparative analysis.

In addition, other exploration-promoting strategies have been examined in RL. The AMIGo approach teaches a goal-generating teacher agent to assign sub-goals to a student agent [24]. A tree-like traversal method is proposed to explore while learning to return to less visited regions [25]. Other strategies involve injecting noise into the parameter space [26] or action space [27] to guide agents to new regions. The Never Give Up (NGU) method combines episodic rewards with global rewards to promote the exploration capabilities of agents [9]. In our work, we manage to enhance some existing intrinsic rewards through within-episode reward adjustments.

The trade-off between exploration and exploitation presents another challenge in reinforcement learning. For simpler tasks, approaches like Bayesian RL [28] and PAC-MDP methods [29], [30] effectively handle this trade-off, offering formal guarantees. However, for more complex tasks, heuristic designs are typically employed [31], [32]. Researchers have investigated the optimal amount of exploration (i.e., the overall exploration-exploitation trade-off) [33], [34], the best actions for exploration [3], [35], [36], and recently, the optimal timing for exploration [31]. Our approach proposes adjusting exploration based on extrinsic rewards to better manage this trade-off.

III. BACKGROUND AND NOTATION

A. MARKOV DECISION PROCESS (MDP)

The scope of our work is within the framework of the Markov Decision Process (MDP), denoted by the tuple $M = \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma$, where $\mathcal{S}$ signifies the state space, $\mathcal{A}$ represents the set of actions, $\mathcal{P}$ is the transition kernel (assumed to be unknown), $\mathcal{R}$ is the reward function, and $\gamma \in [0, 1)$ serves as the discount factor. At each step, the agent interacts with the state space $\mathcal{S}$ by observing a state s and choosing an action $a \in \mathcal{A}$ based on the policy function $\pi(s)$. The state s then transitions to a new state s' with a probability $P(s'|s, a)$, resulting in the agent receiving a reward $r(s, a)$.

B. POLICY GRADIENT

Reinforcement Learning (RL) aims to maximize the agent's expected cumulative return $\mathbb{E}_\pi\left[\sum_{t=0}^{H} \gamma r^t\right]$. This is achieved by taking steps in the direction of the gradient of the expected return with respect to the policy parameters. The expected return, denoted as $J(\theta)$, depends on the policy parameter θ. The policy gradient provides the gradient of $J(\theta)$ as follows:

$$\nabla J(\theta) = \mathbb{E}_\tau\left[\sum_{t=0}^{T} A_t \nabla \log \pi_\theta(a_t|s_t)\right] \tag{1}$$

where A_t represents the advantage function, which measures the advantage of taking a specific action in a particular state over randomly selecting an action according to the policy π_θ. We use the definition of the advantage function as:

$$A_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{2}$$

where the value function $V(s_t)$ of a state s_t represents the expected return from that state when following the policy π_θ.

C. INTRINSIC REWARDS

Various intrinsic reward mechanisms have been explored in the researches [6], [8], [12], [13]. In these works, the agent receives an intrinsic reward, r_t^{i}, at each timestep in addition to an extrinsic reward, r_t^{e}. The learning objective becomes optimizing the weighted sum of intrinsic and extrinsic rewards:

$$r_t = r_t^{\mathrm{e}} + \beta r_t^{\mathrm{i}} \tag{3}$$

where β serves as a balancing coefficient for the two rewards. This reward coefficient plays a significant role in both the efficiency and performance of learning, and these aspects are extensively discussed in this paper.

Episodic state count defines intrinsic rewards based on the frequency of a given state visitation within an episode [13]:

$$r_t^{\mathrm{count}} = \frac{1}{\sqrt{N_{\mathrm{ep}}(s_{t+1})}} \tag{4}$$

where $N_{\mathrm{ep}}(s_{t+1})$ denotes the number of occurrences of state s_t within the current episode.

RIDE generates intrinsic rewards by calculating the distance between the embeddings of two consecutive observations. These rewards are then modulated based on the episodic state count [13]:

$$r_t^{\mathrm{ride}} = ||f_{\mathrm{emb}}(s_t) - f_{\mathrm{emb}}(s_{t+1})||_2 \cdot \frac{1}{\sqrt{N_{\mathrm{ep}}(s_{t+1})}} \tag{5}$$

where $f_{\mathrm{emb}}(s)$ represents a learned state representation as the ICM [6].

AGAC computes intrinsic rewards based on the KL divergence between the agent's policy distribution and an

adversarial policy distribution [12]:

$$r_t^{\text{agac}} = \beta^{\text{adv}} \cdot D_{\text{KL}}(\pi(\cdot|s_t)||\pi_{\text{adv}}(\cdot|s_t)) + \beta^{\text{count}} \cdot \mathbb{I}[N_{\text{ep}}(s_{t+1}) = 1] \quad (6)$$

where β^{adv} is the adversarial bonus coefficient, β^{count} is the count-based bonus coefficient, $\pi(\cdot)$ is the policy function, $\pi_{\text{adv}}(\cdot)$ represents the adversarial policy (trained by mimicking the actions of the policy), and $\mathbb{I}(\cdot)$ is an indicator function.

IV. ADAPTIVE INTRINSIC REWARD REGULATION

A. PROBLEM FORMULATION

The method presented here serves as a framework that enhances the performance of existing intrinsic rewards by incorporating the exploration-exploitation trade-off. This trade-off encompasses two dimensions: one that spans the training process and one that occurs within each episode.

The proposed method replaces the coefficient in Equation (3) with an adaptive coefficient denoted as $\mathcal{C}(n, t)$:

$$r_t = r_t^{\text{e}} + \mathcal{C}(n, t) \cdot r_t^{\text{i}} \quad (7)$$

To adequately adjust the intrinsic rewards, the adaptive factor consists of two parts:

$$\mathcal{C}(n, t) = \mathcal{C}^{\text{ext}}(n) \cdot \mathcal{C}^{\text{ep}}(t) \quad (8)$$

where $\mathcal{C}^{\text{ext}}(n)$ represents the extrinsic regulation coefficient, which is adjusted based on extrinsic rewards throughout the training process, in which n is the episode index, and $\mathcal{C}^{\text{ep}}(t)$ denotes the episodic regulation coefficient, which is adjusted based on the timesteps t within each episode. The details of the design for these two coefficients will be presented next.

B. EXTRINSIC REGULATION COEFFICIENT

The extrinsic regulation coefficient $\mathcal{C}^{\text{ext}}(n)$ is utilized to adjust the intrinsic rewards according to the feedback obtained from extrinsic rewards. This coefficient considers the current extrinsic reward and the maximum extrinsic reward.

$$\mathcal{C}^{\text{ext}}(n) = \frac{\bar{r}^{\text{top}} - \bar{r}^{\text{last}}}{d^{\max}} \quad (9)$$

where $\bar{r}^{\text{top}}$ denotes the average of the top K extrinsic rewards, $\bar{r}^{\text{last}}$ represents the average of the last K extrinsic rewards, $d^{\max}$ signifies the maximum difference $(\bar{r}^{\text{top}} - \bar{r}^{\text{last}})$ between $\bar{r}^{\text{top}}$ and $\bar{r}^{\text{last}}$ since the beginning of training. In the implementation, we use two buffers of length K to calculate $\bar{r}^{\text{top}}$ and $\bar{r}^{\text{last}}$ respectively, as depicted in Figure 1. The buffer introduces a smoothing effect on extrinsic rewards.

The difference between the average return in the buffer and the agent's current return tends to be minimal both at the start of the training and when the training approaches convergence. Consequently, $\mathcal{C}^{\text{ext}}(n)$ will approach a very small value once the algorithm converges.

To prevent significant fluctuations in the coefficient that might shock the training process, we adjust $\mathcal{C}^{\text{ext}}(n)$ incrementally, expressed as $\mathcal{C}^{\text{ext}}(n) = \mathcal{C}^{\text{ext}}(n) + \Delta\mathcal{C}^{\text{ext}}(n)$. We also apply a constraint ϵ to limit the magnitude of each update. The adjusted equation becomes:

$$\mathcal{C}^{\text{ext}}(n) = \mathcal{C}^{\text{ext}}(n) + \text{clip}\Big(\frac{\bar{r}^{\text{top}} - \bar{r}^{\text{last}}}{d^{\max}} - \mathcal{C}^{\text{ext}}(n), -\epsilon,\ 0\Big) \quad (10)$$

The coefficient calculation leverages the changes in the difference between the maximum extrinsic rewards and the current extrinsic rewards. These differences typically exhibit peaks during training, which then decrease after convergence. We utilize these changes to adjust the intrinsic rewards and promote learning efficiency. The heuristic approach involves gradually decreasing the intrinsic rewards coefficient as the extrinsic rewards approach convergence. This reduction aims to minimize the interference of intrinsic rewards on the learning process of the agent, and helps in focusing more on these valuable extrinsic cues.

C. EPISODIC REGULATION COEFFICIENT

The episodic regulation coefficient $\mathcal{C}^{\text{ep}}(t)$ adjusts the ratio of the intrinsic rewards at each timestep within an episode. Here, $\mathcal{C}^{\text{ep}}(t) \in [0, 1]$ can be any monotonically increasing function used for step-based tuning. In this work, we define it as follows:

$$\mathcal{C}^{\text{ep}}(t) = \min(1, \frac{t}{L}) \quad (11)$$

where t represents the timestep in an episode, L is a normalization factor that can be manually specified or derived from the maximum length of collected trajectories. The operator $\min(\cdot)$ ensures that $\mathcal{C}^{\text{ep}}(t)$ remains less than or equal to 1. The choice of parameter L will be discussed in detail in Section V-E.

In trajectory-wise reward tasks, the episodic regulation coefficient is utilized to reduce excessive incentive states near the beginning of an episode, where states are easily observed or frequently visited. It also amplifies the intrinsic rewards near the end of an episode, where states are less likely to be observed, thus necessitating larger intrinsic rewards. On the other hand, as the lengths of episodes decrease, the overall impact of intrinsic rewards diminishes, further reducing their potential interference with the agent's learning process.

D. POLICY GRADIENT WITH REWARD REGULATION

During training, we employ the defined rewards as described in Equation (7). Consequently, we redefine the advantage in the policy gradient loss by substituting Equation (7) into Equation (2). To simplify notation, we replace $\mathcal{C}(n, t)$ with $\mathcal{C}$.

$$\begin{aligned} A_t &= r_t + \gamma V(s_{t+1}) - V(s_t) \\ &= (r_t^{\text{e}} + \mathcal{C} \cdot r_t^{\text{i}}) + \gamma V(s_{t+1}) - V(s_t) \end{aligned} \quad (12)$$

where the value estimator $V(s_t)$ is obtained by minimizing the TD residual $\big((r_t^{\text{e}} + \mathcal{C} \cdot r_t^{\text{i}}) + \gamma V(s_{t+1}) - V(s_t)\big)^2$. However, learning is hindered by the fact that the target $(r_t^{\text{e}} + \mathcal{C} \cdot r_t^{\text{i}}) + \gamma V(s_{t+1})$ changes at each timestep and each episode due to the changing $\mathcal{C}$. To address this issue, we split the value

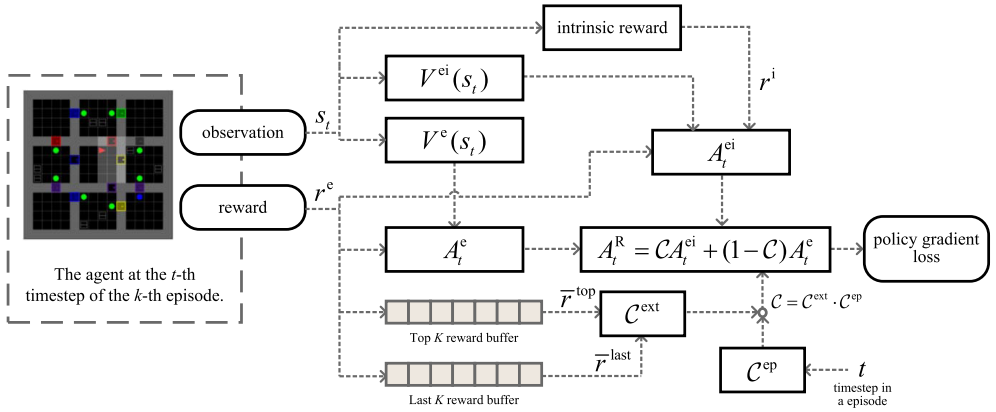


FIGURE 1. The architecture of adaptive intrinsic reward regulation, illustrating the components of the advantage A^R. The components include the extrinsic regulation coefficient C^{ext}, the episodic regulation coefficient C^{ep}, the extrinsic and intrinsic reward advantage A^{ei}, and the extrinsic reward advantage A^e.

estimator $V(s_t)$ into two parts, namely $V^{ei}(s_t)$ and $V^e(s_t)$, to counteract this change:

$$V(s_t) = \mathcal{C}V^{ei}(s_t) + (1-\mathcal{C})V^e(s_t) \tag{13}$$

where $V^{ei}(s_t)$ and $V^e(s_t)$ denote the value estimators that account for both intrinsic rewards and extrinsic rewards, and extrinsic rewards alone, respectively. They are updated by minimizing the following loss:

$$\begin{aligned}\mathcal{L}(V^{ei}) = \mathbb{E}_{(s_t,r^e,r^i,s_{t+1})\sim\tau}\Big[\Big((r_t^e + r_t^i) + \gamma V^{ei}(s_{t+1}) \\ - V^{ei}(s_t)\Big)^2\Big] \\ \mathcal{L}(V^e) = \mathbb{E}_{(s_t,r^e,s_{t+1})\sim\tau}\Big[\Big(r_t^e + \gamma V^e(s_{t+1}) \\ - V^e(s_t)\Big)^2\Big]\end{aligned} \tag{14}$$

Consequently, the advantage A_t can be rewritten using $V^{ei}(s_t)$ and $V^e(s_t)$ to replace $V(s_t)$:

$$\begin{aligned}A_t &= (r_t^e + \mathcal{C}r_t^i) + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) \\ &= \mathcal{C}(r_t^e + r_t^i) + (1-\mathcal{C})r_t^e + \gamma\big(\mathcal{C}V^{ei}(s_{t+1}) \\ &\quad + (1-\mathcal{C})V^e(s_{t+1})\big) - \big(\mathcal{C}V^{ei}(s_t) + (1-\mathcal{C})V^e(s_t)\big) \\ &= \mathcal{C}\big((r_t^e + r_t^i) + V^{ei}(s_t) + V^{ei}(s_t)\big) \\ &\quad + (1-\mathcal{C})\big(r_t^e + V^e(s_t) + V^e(s_t)\big)\end{aligned} \tag{15}$$

For simplicity of notation, we define the extrinsic and intrinsic reward advantages A^{ei} and the extrinsic reward advantage A^e:

$$\begin{aligned}A_t^{ei} &\triangleq (r_t^e + r_t^i) + V^{ei}(s_t) + V^{ei}(s_t) \\ A_t^e &\triangleq r_t^e + V^e(s_t) + V^e(s_t)\end{aligned} \tag{16}$$

Using these definitions, the advantage with reward regulation A_t can be expressed as A_t^R:

$$A_t^R = \mathcal{C}A_t^{ei} + (1-\mathcal{C})A_t^e \tag{17}$$

Algorithm 1 Adaptive Intrinsic Reward Regulation

Input: Training steps S, RL rollout steps T
Initialize the policy π_θ
for batch iteration $n = 1, 2, \ldots$ until N **do**
  Execute π_θ for T timesteps to generate T tuples of $(s_t, a_t, r_t^e, s_{t+1})$
  for each timestep t in generated episode τ **do**
    if $t = 0$ **then**
      Calculate $\mathcal{C}^{ext}(n)$ in Equation (10)
    end if
    Calculate $\mathcal{C}^{ep}(t)$ and $\mathcal{C}(n,t)$ in Equation (11) and Equation (8).
    Compute intrinsic reward r_t^i according to the specific method.
    Compute advantage A_t^R in Equation (17).
  end for
  Update π_θ and value functions, V^{ei} and V^e, in Equation (18) and Equation (14).
end for

The objective of the policy gradient in Equation (1) is then rewritten as:

$$\nabla J(\theta) = \mathbb{E}_\tau\left[\sum_{t=0}^{T}(\mathcal{C}A_t^{ei} + (1-\mathcal{C})A_t^e)\nabla\log\pi_\theta(a_t|s_t)\right] \tag{18}$$

In summary, adaptive intrinsic reward regulation contains two regulation coefficients: the extrinsic regulation coefficient and the episodic regulation coefficient. We modify the calculation of the advantage in the policy gradient loss to address the issue of changing target values during training, which improve learning stability. The proposed method is depicted in Figure 1, and the algorithmic procedure is summarized in Algorithm 1.

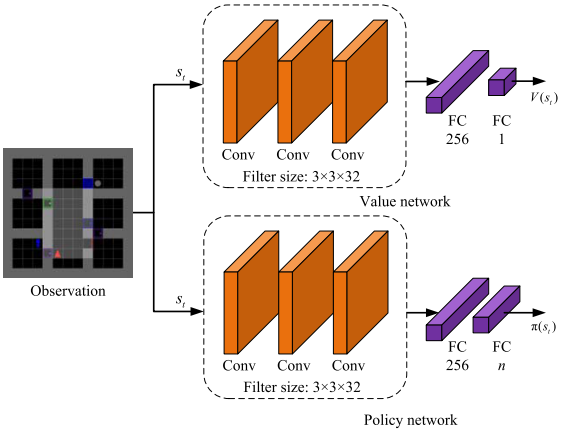


FIGURE 2. The architecture of the neural networks for the policy function and the value function. These two networks do not share parameters and are applied in all MiniGrid, ViZDoom, and MiniWorld experiments.

V. EXPERIMENTS

We conduct experiments to evaluate the performance of Adaptive Intrinsic Reward Regulation (AdaReg) in three environments with discrete actions: MiniGrid, ViZDoom, and MiniWorld, in which MiniGrid and MiniWorld are procedurally-generated environments. In these environments, agents receive rewards only at the end of an episode. We assess the performance of our method from four different perspectives:

1) Comparative performance of our proposed method versus baseline methods in procedurally-generated environments with high reward sparsity.
2) Efficacy of our method in high-dimensional environments.
3) The correlation between the normalization factor L and performance.
4) Individual contribution of each module to the overall performance.

Next, we discuss these points in detail.

A. EXPERIMENTAL DETAILS

To demonstrate the performance enhancements facilitated by AdaReg, we integrate it with four distinct intrinsic reward methodologies. The four baseline methods we used are:

1) The Adversarially Guided Actor-Critic (AGAC) approach, which is an adversarial method [12].
2) The Episodic State Count (COUNT) method, utilized as a count-based exploration technique [7], [11].
3) Rewarding Impact-Driven Exploration (RIDE), designed specifically for procedurally-generated environments [13].
4) Elliptical Episodic Bonuses (E3B), a count-based method suitable for high-dimensional observations [37]. This method is evaluated in a 3D maze environment.

The experimental results, including both our method and the baselines, employ identical neural network architectures for the policy function and value function. The architecture, depicted in Figure 2, consists of three convolutional layers and two fully connected layers, with separate parameters for each network. This architecture is inspired by the designs presented in [12] and [13]. Although alternative architectures may offer potential performance improvements, using a uniform network structure ensures comparability across results. Additionally, RIDE, E3B and AGAC require additional neural networks for their learning functionalities. We follow the designs outlined in the respective literature for RIDE and E3B, while for AGAC, an extra policy network, depicted as the policy network in Figure 2, is employed to learn the adversarial policy.

Two primary hyperparameters require adjustment during experimentation. The first one is the intrinsic reward coefficient, denoted as β. We perform grid searches over the intrinsic reward coefficient, with values $\beta \in [0.3, 0.1, 0.03, 0.01, 0.003, 0.001]$ for the RIDE and COUNT methods in the 2D Maze and 3D Maze environments, respectively. We perform grid searches for E3B with values $\beta \in [10^{-5}, 6 \cdot 10^{-6}, 3 \cdot 10^{-6}, 10^{-6}]$. For AGAC, we utilize the parameter settings as described in [12], in which β^{adv} is the coefficient of adversarial bonus and β^{count} is the coefficient of count-based bonus. The selected values for 2D Maze and 3D Maze are shown in Tables 1 and 2.

The normalization coefficient, denoted as L in Equation (11), is the second adjustable parameter. The maximum episode length, usually provided by the environment simulator, can typically be obtained directly via the simulator interface. However, for the sake of generality, we assume that the maximum length remains unknown and therefore require recording the length of each episode:

$$H_k = H_{k-1} \cup \{h_k\} \tag{19}$$

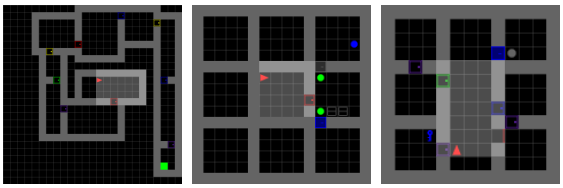
where h_k denotes the length of the k-th episode, H_k collects the lengths of the first k episodes, and $H_0 = \Phi$. The maximum episode length is then derived from $\max(H_k)$.

Furthermore, the buffer length K is set to 100, and ϵ is set to 0.001. The values of K and ϵ have a minor impact on the experimental results. In our experiments, we use these values and will not discuss them further. Parameters associated with specific algorithms, such as the parameters in RIDE, AGAC and E3B, are configured as specified in their researches [12], [13], [37]. The chosen values are reported in Tables 1 and 2.

B. RESULTS ON MINIGRID

We evaluate our approach on various procedurally-generated tasks with different levels of complexity in the MiniGrid environment and compare the outcomes to those of baseline methods. The MiniGrid environment [38] consists of a suite of challenging tasks that are generated procedurally. These tasks are characterized by partial observation and sparse-reward gridworlds. Our focus is on three task types: MultiRoom, KeyCorridor, and ObstructedMaze, which are depicted in Figure 3.

All methods take the current observation and the previous three observations as input to the policy and value functions.



(a) (b) (c)

FIGURE 3. Illustrations of scenarios in the MiniGrid environment: (a) The MultiRoom task involves multiple rooms, where the agent, indicated by the red arrow, must open the door, traverse all rooms, and reach the green square, marking the goal position. (b) In the ObstructedMaze task, the agent needs to open the door, clear obstacles, locate the key within a box, open the subsequent door, and enter the final room to reach the goal, denoted by a ball. (c) In the KeyCorridor task, the agent must find the key in a room, unlock the door to access the final room, and reach the goal, represented by a ball.

TABLE 1. Hyperparameters for MiniGrid experiments.

Method	Hyperparameter	Value
AGAC+AdaReg / AGAC	β^{count}	0.1
AGAC+AdaReg / AGAC	β^{adv}	0.0004
COUNT+AdaReg / COUNT	β	0.03
RIDE+AdaReg / RIDE	β	0.1
RIDE+AdaReg / RIDE	Forward loss coefficient	10
RIDE+AdaReg / RIDE	Inverse loss coefficient	0.1

The inputs are the partial observations of the agent's view with the size of $7 \times 7 \times 3$. The implementation details of the intrinsic reward methods are as follows: For AGAC, we follow the same MiniGrid settings as in the literature [12]. We use partially observable states to generate adversarial bonuses in combination with fully observable state counts. For COUNT, we utilize the fully observable state count as intrinsic rewards, following the configuration in the literature [11]. For RIDE, we use partially observable states to generate intrinsic rewards. The hyperparameter settings for the experiments are presented in Table 1.

Figure 4 shows the performance curves of our proposed method and the baselines. We compare the performance of our method with its baseline on each task. In the ObstructedMaze tasks, AdaReg consistently outperforms the baselines. In the KeyCorridor tasks, our method significantly improves performance. In the MultiRoom tasks, our approach shows a slight advantage. This could be due to the presence of multiple similar scenarios across the rooms, which reduces the effect of episodic regulation. However, extrinsic regulation continues to enable the average return curve to converge to a larger value by reducing intrinsic rewards.

Furthermore, we compare the results of AdaReg integration, including AGAC+AdaReg, COUNT+AdaReg, and RIDE+AdaReg. We find that AGAC+AdaReg achieves similar performance to COUNT+AdaReg. This phenomenon could be due to the fact that both AGAC and COUNT utilize fully observable counts. However, the adversarial bonus within AGAC might dilute the role of counting, leading to a performance gap between the two. The incorporation of AdaReg mitigates this adverse impact, aligning the average return curve of AGAC+AdaReg with that of COUNT+AdaReg. This demonstrates that our proposed method not only amplifies the effects of existing intrinsic rewards but also effectively curbs their potential negative implications.

We have demonstrated the performance improvement of our method compared to COUNT and RIDE, and confirmed the suppression of adverse effects in AGAC. However, further empirical verification is needed to determine whether AdaReg can enhance the effectiveness of the adversarial bonus adjustment in AGAC. This will be addressed in the subsequent two subsections.

TABLE 2. Hyperparameters used in experiments on ViZDoom and MiniWorld.

Method	Hyperparameter	Value
AGAC+AdaReg / AGAC	β^{count}	0
AGAC+AdaReg / AGAC	β^{adv}	0.00004
COUNT+AdaReg / COUNT	β	0.01
RIDE+AdaReg / RIDE	β	0.001
E3B+AdaReg	β	$6 \cdot 10^{-6}$
E3B	β	$3 \cdot 10^{-6}$
RIDE+AdaReg / RIDE	Forward loss coefficient	0.5
RIDE+AdaReg / RIDE	Inverse loss coefficient	0.8

C. RESULTS ON VIZDOOM

We conduct an evaluation of our approach using ViZDoom [39], a 3D maze environment. Specifically, we focus on the MyWayHome task, which requires the agent to navigate through corridors and locate the room containing the final goal. Figure 5 provides an illustration and description of these environments.

All methods utilize the agent's perspective, which is converted to grayscale, resized to 42×42 pixels, and combined with the previous three frames. This composite image serves as the input for both the policy function and the value function. Additionally, AGAC, COUNT, RIDE and E3B use the current agent's perspective as input for intrinsic rewards. For comparison purposes, we include an average return curve for the experiment where the agent is trained without any intrinsic rewards (NoIntReward). The hyperparameter settings can be found in Table 2.

Figure 6 presents the performance curves of our proposed method and its baselines, along with a comparison to the curve excluding intrinsic rewards. The methods incorporating AdaReg demonstrate significant improvements over the baseline method. The outcomes achieved by AGAC and RIDE in the experiment align with those reported in the literature [12], and the result of E3B is consistent with that in the literature [37].

Comparing the results to NoIntReward, it is evident that AGAC and RIDE yield similar performance. This suggests that the MyWayHome task is not particularly challenging, and the agent can learn without relying on intrinsic rewards. This similarity in return curves between RIDE and NoIntReward has also been reported in the literature [11]. On the other

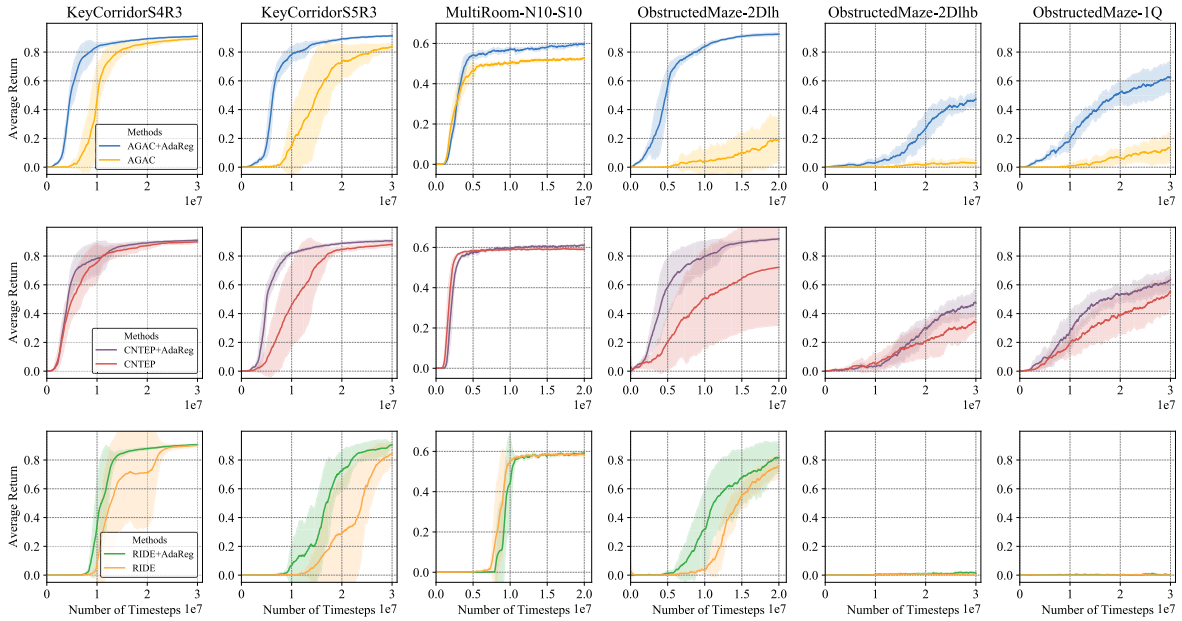


FIGURE 4. **Performance comparison of the AdaReg algorithm in conjunction with AGAC, COUNT, and RIDE within the MiniGrid environment. The first row depicts the performance comparison of AdaReg combined with AGAC. The second row illustrates the performance comparison of AdaReg integrated with COUNT. The third row presents the performance comparison of AdaReg combined with RIDE. Each column corresponds to each task. All the experiments are run five times with random seeds, and the shaded area denotes the standard deviation of returns. The horizontal axis signifies the quantity of agents' timesteps within tasks, and the vertical axis corresponds to the average return acquired by agents.**

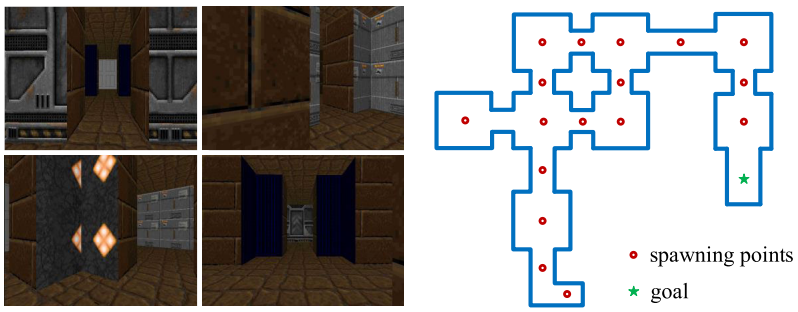


FIGURE 5. **An illustration and brief description of the MyWayHome task within the VizDoom environment. (a) Sample scenarios of the MyWayHome task from the agent's perspective. (b) The structure of the MyWayHome task, which consists of 9 rooms interconnected by corridors. The agent must reach the goal location within 2100 steps, or else the task is considered a failure. The red circle represents the agent's spawn locations, and at the start of each episode, the agent appears randomly at one of these positions. The green star denotes the agent's goal location.**

hand, COUNT, due to the potential for large rewards in high-dimensional environments, rarely improves learning and can sometimes create interference. Interestingly, the method proposed in this paper can mitigate the effects of detrimental intrinsic rewards, leading to more stable learning outcomes. This phenomenon can be attributed to the implementation of extrinsic regulation, which progressively diminishes the impact of intrinsic rewards as the average return curve increases.

Unlike the COUNT method, which can interfere with the agent's learning, the counting approach of E3B is designed for high-dimensional observations and thus facilitates learning in 3D environments. The results demonstrate that the average reward curves for AdaReg+E3B outperform those using E3B alone. The application of AdaReg's extrinsic regulation effectively mitigates performance degradation issues in E3B after convergence, which are often due to the excessive influence of intrinsic rewards.

Our experiments have confirmed the improved performance of our method on AGAC, RIDE and E3B, as well as the alleviation of interference caused by COUNT in high-dimensional environments. However, since VizDoom is a fixed 3D maze environment, we aim to further validate our work in a procedurally-generated 3D maze environment.

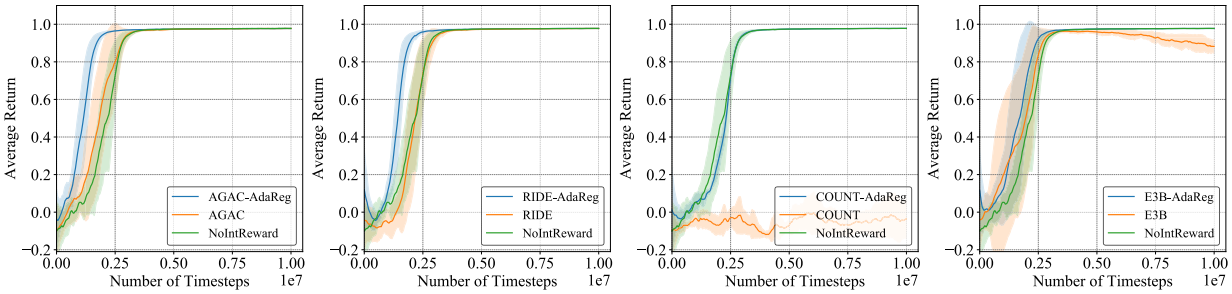


FIGURE 6. A comparative performance assessment of the AdaReg algorithm integrated with AGAC, COUNT, RIDE and E3B on the MyWayHome task of the VizDoom environment, with NoIntReward serving as a baseline. **First:** Performance comparison of AdaReg combined with AGAC. **Second:** Performance comparison of AdaReg combined with RIDE. **Third:** Performance comparison of AdaReg combined with COUNT. **Fourth:** Performance comparison of AdaReg combined with E3B. All experiments are run five times with random seeds, and the shaded area denotes the standard deviation of returns. The horizontal axis corresponds to the agents' timesteps in tasks, while the vertical axis represents the average return acquired by the agents.

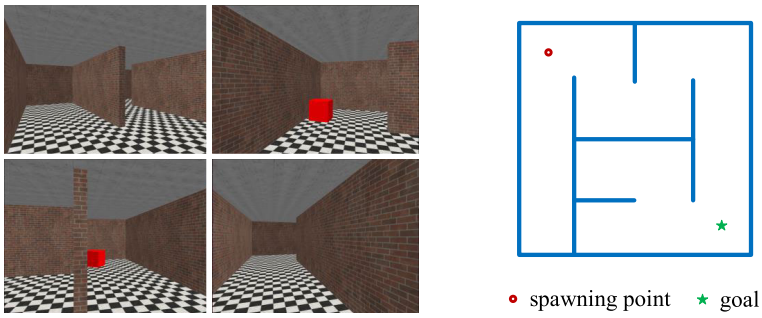


FIGURE 7. Illustration and description of the MazeS4 task within the MiniWorld environment. (a) Exemplar scenarios of the MazeS4 task as seen by the agent. (b) Representative structure of the MazeS4 task, procedurally generated for each episode. The maze is a 4 × 4 grid, and the agent must navigate from the spawn point to the goal within a maximum of 384 steps; failure to do so results in an unsuccessful episode. The red circle represents the agent's spawn point, while the green star denotes the goal.

D. RESULTS ON MINIWORLD

Experiments are conducted using the MazeS4 task within the MiniWorld environment [39], which consists of a procedurally-generated 4 × 4 maze. The implementation of Maze tasks in [15] is used for these experiments, deviating from the official version in [40]. In the official implementation of maze tasks, starting and goal positions are randomly placed on these maze grids. However, the implementation in [15] maintains constant starting and goal positions. In the modified maze task, agents always appear at the top left of the map, and the goal position is fixed at the bottom right of the map. Figure 7 provides an illustrative overview of the environment.

All methods used in the experiments rely on the agent's perspective, which is resized to 42 × 42 × 3 pixels. This processed image serves as the input for the policy and value functions. AGAC, COUNT, RIDE and E3B utilize the current view of the agent to generate intrinsic rewards. Additionally, the experiments include an average return curve of NoIntReward, which is an agent trained without intrinsic rewards, for the purpose of comparison. The hyperparameters employed in the experiment are detailed in Table 2.

In Figure 7, we present performance curves of the proposed method alongside its baseline and compare it with the case without intrinsic rewards. The proposed method demonstrates an enhancement compared to the baseline. However, the procedurally-generated environment implies that the scenarios differ across episodes, making certain intrinsic reward methodologies, such as AGAC and COUNT, ineffective in achieving the same level of learning performance as extrinsic rewards alone. In contrast, RIDE is specifically tailored for procedurally-generated environments, which accounts for its markedly superior performance relative to NoIntReward. After combining RIDE with the AdaReg method, there is an observable improvement in performance. Similar to results observed in the VizDoom environment, the combination of E3B with AdaReg enhances its effectiveness and reduces performance deterioration during the algorithm's convergence phase.

Furthermore, when comparing these experimental outcomes with NoIntReward, it is evident that the return curves for AGAC and COUNT are inferior to NoIntReward. This suggests that the AGAC's adversarial bonuses hamper the agent's exploration within the dynamic environment, and

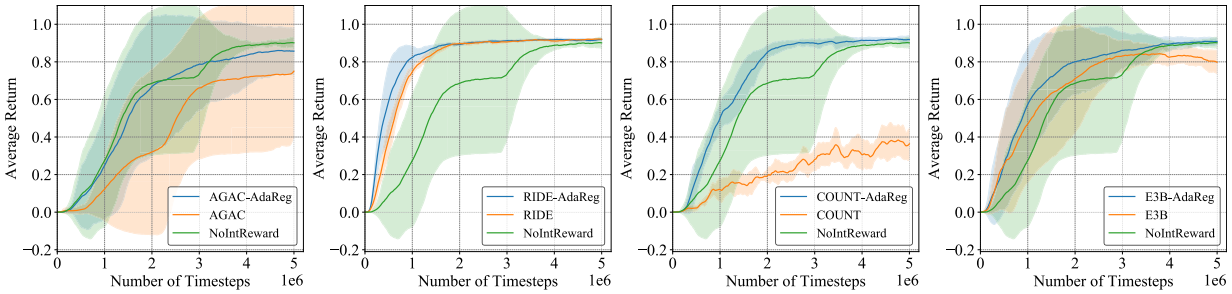


FIGURE 8. Comparison of performance among the AdaReg algorithm integrated with AGAC, COUNT, RIDE and E3B in the MazeS4 task within the MiniWorld environment, using NoIntReward as a baseline. **First:** Performance of AdaReg when combined with AGAC. **Second:** Performance of AdaReg in conjunction with RIDE. **Thrid:** Performance of AdaReg coupled with COUNT. **Fourth:** Performance of AdaReg coupled with E3B. All experiments are run five times with random seeds, and the shaded area denotes the standard deviation of returns. The horizontal axis corresponds to the number of timesteps taken by agents within the tasks, while the vertical axis represents the average return accrued by the agents.

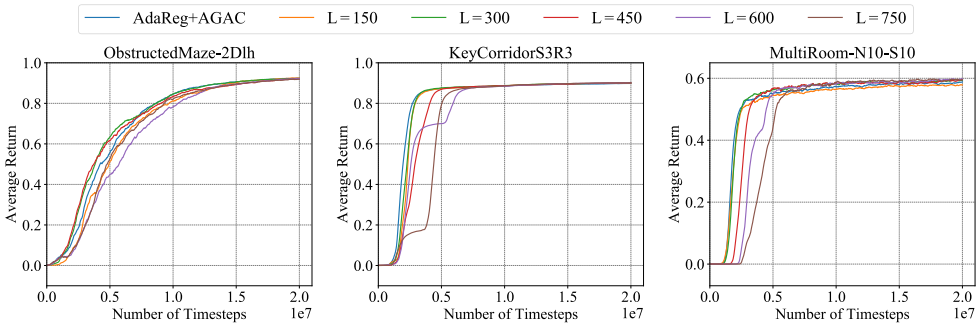


FIGURE 9. Performance of AdaReg with various normalization factor settings. The normalization factor L is evaluated with AdaReg, $L = 150$, $L = 300$, $L = 450$, $L = 600$, and $L = 750$. Each experiment is run five times with random seeds. For clarity in the graph, only the average curve is drawn.

COUNT generates a large number of rewards in high-dimensional environments, which impacts the learning of the agent. The episodic regulation in AdaReg can temper the influence of intrinsic rewards in accordance with extrinsic rewards, thereby mitigating the disruptive effect of intrinsic rewards and enabling the agent to generate an improved average return curve.

E. EFFECTS OF THE NORMALIZATION FACTOR

In the preceding experiments, we consistently observe optimal outcomes when using the maximum trajectory length as the normalization factor. However, in certain tasks, determining the maximum trajectory length may not always be feasible. Therefore, it is important to evaluate the performance of the method when the normalization factor is assigned manually and compare these results with those obtained from using the maximum trajectory length.

The comprehensive results are depicted in Figure 9. In this experiment, we select three distinct tasks for comparison: ObstructedMaze-2Dlh, KeyCorridorS3R3, and MultiRoom-N10-S10. These tasks vary in difficulty, with ObstructedMaze-2Dlh being more challenging than KeyCorridorS3R3, which, in turn, is harder than MultiRoom-N10-S10. Additionally, each task allows a different maximum number of steps, as specified in Table 3.

In these three environments, we test five manually specified values of L in Equation (11) and compare them with the results of the proposed method with AGAC (AdaReg+AGAC). In ObstructedMaze-2Dlh, the results for each parameter are closely clustered, with only $L = 150$ causing a noticeable degradation in performance. In KeyCorridorS3R3, the outcomes of three experiments are very similar, while the other three results ($L = 450$, $L = 600$, and $L = 750$) deviate more significantly from the maximum trajectory length, resulting in a more pronounced performance loss. Similarly, in MultiRoom-N10-S10, performance drops considerably as L diverges from the maximum trajectory length.

These findings suggest that even if the maximum trajectory length is not accurately determined, manually specifying the normalization factor can still yield a performance improvement. Thus, it can be inferred that our proposed method demonstrates robustness towards the selection of L.

TABLE 3. Maximum allowable steps in tasks.

Task	Max Steps
ObstructedMaze-2Dlh	576
KeyCorridorS3R3	270
MultiRoom-N10-S10	200

F. ABLATIONS

The proposed method consists of two main elements: extrinsic regulation and episodic regulation. To determine

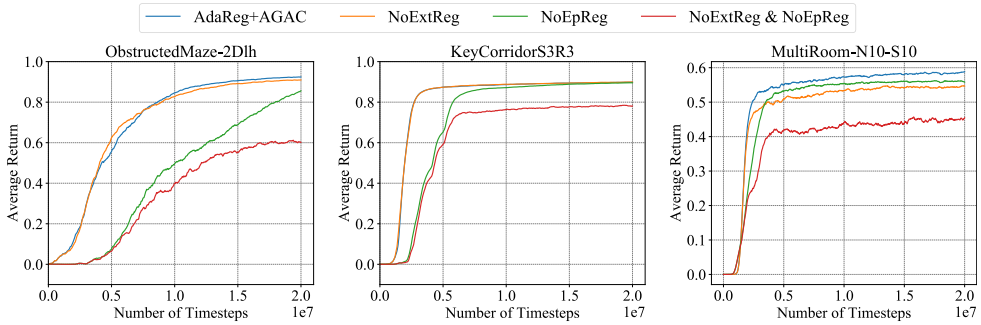


FIGURE 10. Results of the ablation study under various settings: AGAC integrated with the proposed method (AdaReg+AGAC), AdaReg+AGAC without extrinsic regulation (NoExtReg), AdaReg+AGAC without episodic regulation (NoEpReg), AdaReg+AGAC without both extrinsic regulation and episodic regulation (NoExtReg & NoEpReg). Each experiment is run five times with random seeds. For clarity in the graph, only the average curve is shown.

the individual contributions of these components, we conduct an ablation study. We examine the influence of the extrinsic regulation coefficient and the episodic regulation coefficient on the learning performance by fixing one or two coefficients. The value of the irrelevance coefficient is fixed at 1 to disregard its influence.

Figure 10 illustrates the performance assessment under four conditions: the proposed method with AGAC (AdaReg+AGAC), AdaReg+AGAC without extrinsic regulation (NoExtReg) where $\mathcal{C}^{\text{ext}} = 1$, AdaReg+AGAC without episodic regulation (NoEpReg) where $\mathcal{C}^{\text{ep}} = 1$, and AdaReg+AGAC without both extrinsic and episodic regulation (NoExtReg & NoEpReg), which is equivalent to $\mathcal{C} = 1$ in Equation (17).

The experiment involves three tasks in MiniGrid: ObstructedMaze-2Dlh, KeyCorridorS3R3, and MultiRoom-N10-S10, representing three levels of difficulty: high, medium, and low, respectively. AdaReg without extrinsic regulation shows slightly inferior performance, followed by AdaReg without episodic regulation. In ObstructedMaze-2Dlh and KeyCorridorS3R3, NoExtReg exhibits similar performance to AdaReg. Particularly in KeyCorridorS3R3, the two curves coincide. In ObstructedMaze-2Dlh, AdaReg converges to a higher value than NoExtReg. In MultiRoom-N10-S10, NoExtReg exhibits its superiority over AdaReg. In these tasks, NoEpReg demonstrates the ability of episodic regulation to improve performance.

The ablation study clearly demonstrates that in AdaReg, episodic regulation is crucial in enhancing performance, while extrinsic regulation facilitates better convergence. Conversely, if both components are removed simultaneously, performance declines significantly. The above results are also consistent with the judgment based on the intuitive analysis.

VI. CONCLUSION

This paper has introduced a novel approach aimed at enhancing intrinsic reward methods in reinforcement learning (RL). The method improves learning efficiency and performance of agents by addressing the challenge of exploration-exploitation trade-off, particularly in environments with trajectory-wise rewards. This is achieved via two regulation coefficients, the extrinsic and episodic regulation coefficients, which adjust intrinsic rewards based on extrinsic feedback and episode timesteps respectively. The method has been demonstrated to provide significant improvements in the performance of RL agents, by successful experiments in diverse environments such as MiniGrid, VizDoom, and MiniWorld. The results of this study validate the proposed method and confirm the improvements it offers to the field of reinforcement learning.

Nevertheless, this study has its limitations. One notable limitation is the absence of a well-defined theoretical framework that outlines the scope of applicability for our method. In essence, while our approach demonstrates effectiveness when combined with specific methods for particular tasks, its applicability may not extend to all environments.

In future research, our objectives are twofold. Firstly, we aim to identify a more universally applicable technique that can enhance existing intrinsic reward methods across a broader spectrum of RL scenarios. Simultaneously, we endeavor to define the boundaries of its applicability or establish guiding principles for selecting suitable environments and tasks where our approach can deliver optimal results.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[2] T. Zhang, P. Rashidinejad, J. Jiao, Y. Tian, J. E. Gonzalez, and S. Russell, "MADE: Exploration via maximizing deviation from explored regions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9663–9680.

[3] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Trans. Evol. Comput.*, vol. 11, no. 2, pp. 265–286, Apr. 2007.

[4] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 2850–2869.

[5] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *J. Mach. Learn. Res.*, vol. 22, no. 98, pp. 1–76, 2021.

[6] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 488–489.

[7] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos, "Unifying count-based exploration and intrinsic motivation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1471–1479.

[8] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," in *Proc. 7th Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[9] A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, and C. Blundell, "Never give up: Learning directed exploration strategies," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–26.

[10] D. Pathak, D. Gandhi, and A. Gupta, "Self-supervised exploration via disagreement," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5062–5071.

[11] K. Wang, K. Zhou, B. Kang, J. Feng, and Y. Shuicheng, "Revisiting intrinsic reward for exploration in procedurally generated environments," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–20.

[12] Y. Flet-Berliac, J. Ferret, O. Pietquin, P. Preux, and M. Geist, "Adversarially guided actor-critic," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–18.

[13] R. Raileanu and T. Rocktäschel, "RIDE: Rewarding impact-driven exploration for procedurally-generated environments," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–21.

[14] C. Dann, M. Mohri, T. Zhang, and J. Zimmert, "A provably efficient model-free posterior sampling method for episodic reinforcement learning," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 12040–12051.

[15] D. Zha, W. Ma, L. Yuan, X. Hu, and J. Liu, "Rank the episodes: A simple approach for exploration in procedurally-generated environments," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–25.

[16] P.-Y. Oudeyer and F. Kaplan, "How can we define intrinsic motivation?" in *Proc. 8th Int. Conf. Epigenetic Robot., Modelling Cogn. Develop. Robotic Syst.*, 2008, pp. 1–9.

[17] A. L. Strehl and M. L. Littman, "An analysis of model-based interval estimation for Markov decision processes," *J. Comput. Syst. Sci.*, vol. 74, no. 8, pp. 1309–1331, Dec. 2008.

[18] G. Ostrovski, M. G. Bellemare, A. Oord, and R. Munos, "Count-based exploration with neural density models," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2721–2730.

[19] L. Kocsis and C. Szepesvári, "Bandit based Monte–Carlo planning," in *Proc. Eur. Conf. Mach. Learn.*, 2006, pp. 282–293.

[20] B. C. Stadie, S. Levine, and P. Abbeel, "Incentivizing exploration in reinforcement learning with deep predictive models," 2015, *arXiv:1507.00814*.

[21] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.J. Ho and S. Ermon, "Generative adversarial imitation learning," in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 1–9.

[22] D. Bahdanau, F. Hill, J. Leike, E. Hughes, A. Hosseini, P. Kohli, and E. Grefenstette, "Learning to understand goal specifications by modelling reward," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

[23] C. Florensa, D. Held, X. Geng, and P. Abbeel, "Automatic goal generation for reinforcement learning agents," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1515–1528.

[24] A. Campero, R. Raileanu, H. Kuttler, J. B. Tenenbaum, T. Rocktäschel, and E. Grefenstette, "Learning with amigo: Adversarially motivated intrinsic goals," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–18.

[25] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune, "Go-explore: A new approach for hard-exploration problems," 2019, *arXiv:1901.10995*.

[26] I. Osband, B. Van Roy, D. J. Russo, and Z. Wen, "Deep exploration via randomized value functions," *J. Mach. Learn. Res.*, vol. 20, no. 124, pp. 1–62, 2019.

[27] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–18.

[28] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, "Bayesian reinforcement learning: A survey," *Found. Trends Mach. Learn.*, vol. 8, nos. 5–6, pp. 359–483, 2015.

[29] P. Auer, "Using confidence bounds for exploitation-exploration tradeoffs," *J. Mach. Learn. Res.*, vol. 3, pp. 397–422, Nov. 2002.

[30] J. Pazis and R. Parr, "Pac optimal exploration in continuous space Markov decision processes," in *Proc. AAAI Conf. Artif. Intell.*, 2013, vol. 27, no. 1, pp. 774–781.

[31] M. Pislar, D. Szepesvari, G. Ostrovski, D. L. Borsa, and T. Schaul, "When should agents explore?" in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–23.

[32] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "VIME: Variational information maximizing exploration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1117–1125.

[33] P. Auer, T. Jaksch, and R. Ortner, "Near-optimal regret bounds for reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21, 2008, pp. 1563–1600.

[34] O. Cappé, A. Garivier, O.-A. Maillard, R. Munos, and G. Stoltz, "Kullback–Leibler upper confidence bounds for optimal sequential allocation," *Ann. Statist.*, vol. 41, no. 3, pp. 1516–1541, Jun. 2013.

[35] J. Schmidhuber, "Curious model-building control systems," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Nov. 1991, pp. 1458–1463.

[36] C. Linke, N. M. Ady, M. White, T. Degris, and A. White, "Adapting behavior via intrinsic reward: A survey and empirical study," *J. Artif. Intell. Res.*, vol. 69, pp. 1287–1332, Dec. 2020.

[37] M. Henaff, R. Raileanu, M. Jiang, and T. Rocktäschel, "Exploration via elliptical episodic bonuses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 37631–37646.

[38] M. Chevalier-Boisvert, L. Willems, and S. Pal. (2018). *Minimalistic Gridworld Environment for OpenAI Gym*. [Online]. Available: https://github.com/maximecb/gym-minigrid

[39] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski, "ViZDoom: A doom-based AI research platform for visual reinforcement learning," in *Proc. IEEE Conf. Comput. Intell. Games (CIG)*, Sep. 2016, pp. 1–8.

[40] M. Chevalier-Boisvert. (2018). *Miniworld: Minimalistic 3D Environment for RL & Robotics Research*. [Online]. Available: https://github.com/maximecb/gym-miniworld

QIAN ZHAO (Member, IEEE) received the M.S. degree from Sichuan University, Chengdu, China, in 2011, and the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, in 2017.

He is currently a Lecturer with the School of Physics and Telecommunication Engineering, Zhoukou Normal University, Zhoukou, China. His current research interests include image processing, computer vision, and machine intelligence.

JINHUI HAN received the Ph.D. degree in electrical circuit and system from the Huazhong University of Science and Technology, Wuhan, China, in 2016.

He is currently an Associate Professor with the School of Physics and Telecommunication Engineering, Zhoukou Normal University, Zhoukou, China. His current research interests include target detection and image processing.

MAO XU received the B.Sc. degree in electronic information engineering from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2012, and the M.Sc. degree in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 2015.

He is currently a Lecturer with the School of Physics and Telecommunication Engineering, Zhoukou Normal University, Zhoukou, China. His current research interests include image processing, computer vision, and machine intelligence.