REWARDING EPISODIC VISITATION DISCREPANCY FOR EXPLORATION IN REINFORCEMENT LEARNING

Mingqi Yuan[1], Bo Li[1], Xin Jin[2] & Wenjun Zeng[2]
[1]Department of Computing, The Hong Kong Polytechnic University
[2]Eastern Institute for Advanced Study

ABSTRACT

Exploration is critical for deep reinforcement learning in complex environments with high-dimensional observations and sparse rewards. To address this problem, recent approaches propose to leverage intrinsic rewards to improve exploration, such as novelty-based exploration and prediction-based exploration. However, many intrinsic reward modules require sophisticated structures and representation learning, resulting in prohibitive computational complexity and unstable performance. In this paper, we propose Rewarding Episodic Visitation Discrepancy (REVD), a computation-efficient and quantified exploration method. More specifically, REVD provides intrinsic rewards by evaluating the Rényi divergence-based visitation discrepancy between episodes. To estimate the divergence efficiently, a k-nearest neighbor estimator is utilized with a randomly-initialized state encoder. Finally, the REVD is tested on Atari games and PyBullet Robotics Environments. Extensive experiments demonstrate that REVD can significantly improve the sample efficiency of reinforcement learning algorithms and outperform the benchmarking methods.

1 INTRODUCTION

Efficient and complete exploration remains one of the significant challenges in deep reinforcement learning. Inspired by human learning patterns, recent approaches propose to leverage intrinsic motivation to improve exploration, especially when the agent receives sparse or missing rewards in real-world scenarios (Schmidhuber, 1991; Oudeyer & Kaplan, 2008; Oudeyer et al., 2007). Intrinsic rewards encourage the agent to visit novel states (Bellemare et al., 2016; Ostrovski et al., 2017; Tang et al., 2017; Burda et al., 2019b) or to increase its knowledge and prediction accuracy of the environment dynamics (Stadie et al., 2015; Pathak et al., 2017; Yu et al., 2020; Houthooft et al., 2016; Oh et al., 2015). For instance, Ostrovski et al. (2017); Bellemare et al. (2016) leveraged a density model to evaluate the novelty of states, and the intrinsic rewards are defined to be inversely proportional to the visiting frequencies. As a result, the agent is motivated to revisit the infrequently-seen states and realize comprehensive exploration. In contrast, Stadie et al. (2015); Pathak et al. (2017); Yu et al. (2020) designed an auxiliary model to learn the dynamics of the environment, and the prediction error is utilized as the intrinsic reward. Furthermore, Burda et al. (2019a) performed experiments on Atari games only using the prediction-based intrinsic rewards, demonstrating that the agent could also achieve remarkable performance.

However, the aforementioned methods cannot provide sustainable exploration incentives because the exploration bonus will vanish as the agent gets to know the environment. As a result, the agent may miss the downstream learning opportunities and fail to learn in hard-exploration tasks. To address the problem, Badia et al. (2020) proposed a never-give-up (NGU) framework that produces intrinsic rewards composed of episodic and long-term state novelty. The episodic state novelty is evaluated using an episodic memory and pseudo-count method, encouraging the agent to visit diverse states in each episode. Since the memory is erased at the beginning of each episode, the intrinsic reward will not vanish across the episodes. In contrast, Raileanu & Rocktäschel (2020) proposed a more straightforward method entitled rewarding-impact-driven-exploration (RIDE). RIDE takes the difference between two consecutive encoded states as the intrinsic reward, encouraging the agent to choose actions that result in large state changes. However, NGU and RIDE are expensive and un-

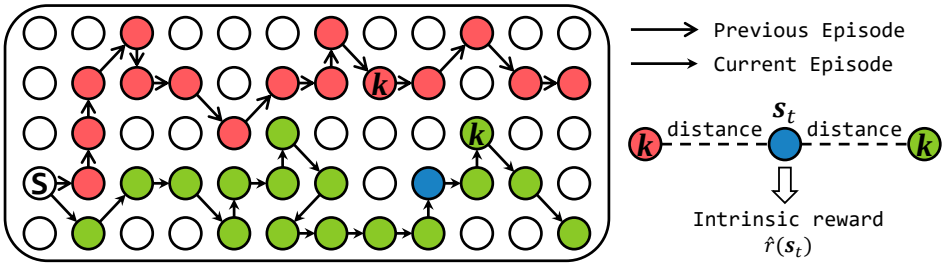


Figure 1: The overview of REVD. Here, **S** represents the starting point, node with a label $\boldsymbol{k}$ represents the k-nearest neighbor (k-NN) of $\boldsymbol{s}_t$ among the previous episode and the current episode. REVD computes the intrinsic reward using the two distances between $\boldsymbol{s}_t$ and its k-NN states.

stable because they rely heavily on auxiliary models. Poor representation learning may incur severe performance loss, making it difficult to be applied to arbitrary tasks.

In this paper, we propose a novel intrinsic reward method entitled **R**ewarding **E**pisodic **V**isitation **D**iscrepancy (REVD), which provides powerful exploration incentives with simple, computation-efficient, and robust architecture. Our main contributions can be summarized as follows:

- REVD requires no memory model or database, which overcomes the problem of vanishing intrinsic rewards and provides sustainable exploration bonuses;
- REVD significantly promotes the sample efficiency of reinforcement learning algorithms without introducing any representation learning and additional models;
- REVD provides a quantified and intuitive metric to evaluate the exploration change between episodes;
- Extensive experiments are performed to compare the performance of REVD against existing methods using Atari games and PyBullet Robotics Environments. Simulation results demonstrate that REVD achieves remarkable performance and outperforms the benchmarking schemes.

2 Related Work

2.1 Novelty-Based Exploration

Novelty-based exploration methods encourage the agent to explore novel states or state-action pairs. Strehl & Littman (2008) utilized the state visitation counts as the exploration bonus in tabular setting and provided theoretical justification for its effectiveness. Bellemare et al. (2016); Machado et al. (2020); Martin et al. (2017); Bellemare et al. (2016); Ostrovski et al. (2017); Tang et al. (2017); Burda et al. (2019b) extended this method to environments with high-dimensional observations. More specifically, Burda et al. (2019b) employed a predictor network and target network (fixed and randomly-initialized) to record the visited states and allocated higher rewards to infrequently-seen states. Martin et al. (2017) proposed a ϕ-pseudo count method by exploiting the feature space of value function approximation, which can estimate the uncertainty related to any state. Machado et al. (2020) designed the exploration bonus using the norm of the successor representation of states, and improved the exploration by capturing the diffusion properties of the environment. However, these novelty-based approaches suffer from vanishing intrinsic rewards and cannot maintain exploration across episodes. Our method can effectively address this problem because it requires no additional memory models or database.

2.2 Prediction-Based Exploration

Prediction-based exploration methods encourage the agent to take actions that reduce the uncertainty of predicting the consequence of its own actions (Stadie et al., 2015; Pathak et al., 2017; Burda et al., 2019a; Yu et al., 2020). Pathak et al. (2017) proposed an intrinsic-curiosity-module (ICM) framework, which uses an inverse-forward pattern to learn a representation space of state and defined the

intrinsic reward as the error of predicting the encoded next-state. A similar representation learning technique is also leveraged in RIDE (Raileanu & Rocktäschel, 2020). In particular, Yu et al. (2020) introduced the intrinsic reward concept to imitation learning and redesigned the ICM using the variational auto-encoder (Kingma & Welling, 2014). However, prediction-based approaches also suffer from vanishing intrinsic rewards and sophisticated representation learning, making it expensive and unstable. Our method follows (Seo et al., 2021) and takes a randomly-initialized neural network to encode the state space, which is cheap and robust to be applied to arbitrary tasks.

2.3 Computation-Efficient Exploration

Computation-efficient exploration aims to provide intrinsic rewards in a cheap, robust, and generalized pattern (Seo et al., 2021; Mutti et al., 2021; Yuan et al., 2022). Seo et al. (2021) suggested maximizing the state entropy and designed a random-encoder-for-efficient-exploration (RE3) method, which uses a k-NN estimator to make efficient entropy estimation and divides the sample mean into particle-based intrinsic rewards. RE3 can significantly promote the sample efficiency of model-based and model-free reinforcement learning algorithms without any representation learning. However, Zhang et al. (2021a) indicated that maximizing the Shannon entropy may lead to a policy that visits some state-action pairs with a vanishing probability and proposed to maximize the Rényi entropy of the state-action distribution (MaxRényi). Yuan et al. (2022) combined the RE3 and MaxRényi by introducing the k-NN estimator and proposed a Rényi state entropy maximization method, and discussed k-value selection and state space encoding in detail. However, Póczos & Schneider (2011) found that the k-NN estimators employed in their work may lead to severe estimation errors because of the inappropriate use of the reverse Fatou lemma. In this paper, we leverage an improved k-NN estimator proposed by (Póczos & Schneider, 2011) to make an efficient divergence estimation. Our method also provides an intuitive and quantified metric to reflect the exploration change between episodes. A closely related work to us is MADE proposed by (Zhang et al., 2021b) that maximizes the deviation from the explored regions. However, MADE needs to estimate the visitation density over state-action pairs, which is non-trivial in complex environments with high-dimensional observations. Therefore, our method is more efficient and can be easily deployed in arbitrary tasks.

3 Preliminaries

3.1 Problem Formulation

In this paper, we study the reinforcement learning problem considering a single agent Markov decision process (MDP), which can be defined as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, \check{r}, \varsigma(\boldsymbol{s}_0), \gamma \rangle$ (Bellman, 1957). Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$ is the state-transition function, where $\Pi(\mathcal{S})$ is the space of probability distributions over $\mathcal{S}$. In addition, $\check{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the extrinsic reward function, $\varsigma(\boldsymbol{s}_0)$ is the initial state distribution, and $\gamma \in [0, 1)$ is a discount factor. In the sequel, we denote the intrinsic reward by $\hat{r}$ to distinguish it from the extrinsic reward. Denote by $\pi(\boldsymbol{a}|\boldsymbol{s})$ the policy of the agent, the objective of reinforcement learning is to learn a policy that maximizes the expected discounted return $R_t = \mathbb{E}_\pi \left[\sum_{k=0}^{T} \gamma^k \check{r}_{t+k+1} \right]$.

3.2 Rényi Divergence

In this paper, we introduce Rényi divergence to evaluate the visitation discrepancy between different episodes, which is a generalized and flexible distance measure. In particular, Rényi divergence of order 1 equals Kullback–Leibler divergence and Rényi divergence of order 0.5 is a function of the square Hellinger distance (Van Erven & Harremos, 2014). Let $X \in \mathbb{R}^d$ be a random vector that has two density functions $p(\boldsymbol{x}), q(\boldsymbol{x})$ with respect to Lebesgue measure on $\mathbb{R}^d$, and let $\{\boldsymbol{x} \in \mathbb{R}^d : p(\boldsymbol{x}), q(\boldsymbol{x}) > 0\}$ be the support set. The Rényi divergence of order $\alpha \in \mathbb{R} \setminus \{1\}$ is defined as

$$D_\alpha(p\|q) = \frac{1}{\alpha - 1} \log \int_{\mathcal{X}} p^\alpha(\boldsymbol{x}) q^{1-\alpha}(\boldsymbol{x}) d\boldsymbol{x}. \tag{1}$$

Since it is non-trivial to evaluate the metric when p and q are unavailable, the following k-NN estimator can be employed for efficient estimation. Let $\mathcal{X} = \{X_i\}_{i=1}^N$ be i.i.d samples from the

distribution with density p and $\mathcal{Y} = \{Y_i\}_{i=1}^{M}$ be i.i.d samples from the distribution with density q, we estimate the Rényi divergence as follows (Póczos & Schneider, 2011; Kandasamy et al., 2015):

$$\hat{D}_{\alpha}^{k,N,M}(p\|q) = \frac{1}{\alpha-1}\log\left\{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{(N-1)\mu_k(X_i,\mathcal{X})}{M\nu_k(X_i,\mathcal{Y})}\right]^{1-\alpha} C_{k,\alpha}\right\}, \tag{2}$$

where $\mu_k(X_i,\mathcal{X})$ is the Euclidean distance between X_i and its k-th nearest neighbor in $\mathcal{X}$, $\nu_k(X_i,\mathcal{Y})$ is the Euclidean distance between X_i and its k-th nearest neighbor in $\mathcal{Y}$, $C_{k,\alpha} = \frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}$, $k > |\alpha-1|$, and $\Gamma(\cdot)$ is the Gamma function.

4 REVD

In this section, we present the REVD: **R**ewarding **E**pisodic **V**isitation **D**iscrepancy, a computation-efficient framework for providing powerful and sustainable exploration incentives in reinforcement learning. The key insight of REVD is to encourage exploration by providing intrinsic rewards based on the episodic visitation discrepancy. Consider the on-policy learning scenario in which the policy is updated at the end of each episode, and let $\tilde{\pi}$ (the former one) and π denote the learned policies of two consecutive episodes. The episodic visitation discrepancy can be evaluated using the Rényi divergence:

$$D_{\alpha}(\rho^{\pi}(\boldsymbol{s})\|\rho^{\tilde{\pi}}(\boldsymbol{s})), \tag{3}$$

where ρ^{π} is the state distribution induced by the policy π. Therefore, if the agent finds novel states as compared to the previous episode, it will lead to a large divergence. This metric reflects the change of episode-level exploration status. In the exploration phase of reinforcement learning, the agent is expected to visit as many distinct states as possible within limited iterations. It is feasible to encourage exploration by maximizing $D_{\alpha}(\rho^{\pi}\|\rho^{\tilde{\pi}})$.

4.1 Intrinsic Reward Design

Next, we transform Eq. (3) into computable intrinsic rewards. Denote by $\{\tilde{\boldsymbol{s}}_i\}_{i=1}^{T}$ and $\{\boldsymbol{s}_i\}_{i=1}^{T}$ the visited states sets of two consecutive episodes, the Eq. (3) can be approximated using Eq. (2) as

$$\begin{aligned}\hat{D}_{\alpha}^{k,T}(\rho^{\pi}\|\rho^{\tilde{\pi}}) &= \frac{1}{\alpha-1}\log\left\{\frac{1}{T}\sum_{i=1}^{T}\left[\frac{(T-1)\mu_k(\boldsymbol{e}_i,\mathcal{E})}{T\nu_k(\boldsymbol{e}_i,\tilde{\mathcal{E}})}\right]^{1-\alpha} C_{k,\alpha}\right\} \\ &\propto \frac{1}{T}\sum_{i=1}^{T}\left[\frac{\nu_k(\boldsymbol{e}_i,\tilde{\mathcal{E}})}{\mu_k(\boldsymbol{e}_i,\mathcal{E})}\right]^{1-\alpha},\end{aligned} \tag{4}$$

where $\boldsymbol{e}_i = f_{\boldsymbol{\phi}}(\boldsymbol{s}_i)$ is an encoding vector produced by a fixed and randomly-initialized encoder, $\mathcal{E} = \{f_{\boldsymbol{\phi}}(\boldsymbol{s}_i)\}_{i=1}^{T}$, and $\alpha \in (0,1)$. Therefore, we define the intrinsic reward that takes each transition as a particle:

$$\hat{r}(\boldsymbol{s}_i) = \left[\nu_k(\boldsymbol{e}_i,\tilde{\mathcal{E}})/(\mu_k(\boldsymbol{e}_i,\mathcal{E})+\epsilon)\right]^{1-\alpha}, \tag{5}$$

where ϵ is a fixed and small constant. A critical problem of this formulation is that the agent may obtain more intrinsic rewards by lingering in a small area. To address the problem, we consider reformulating the intrinsic reward as

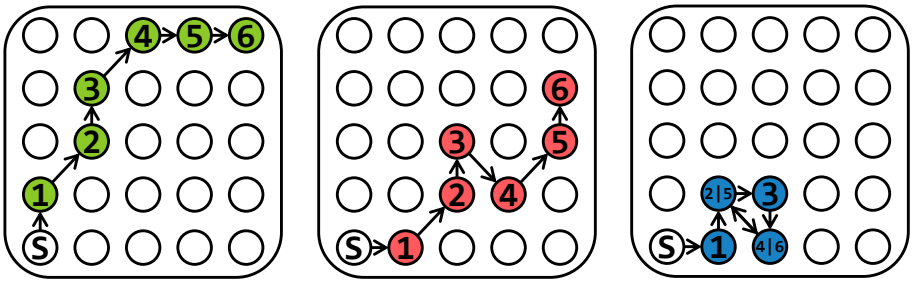
$$\hat{r}(\boldsymbol{s}_i) = L(\mathcal{E})\cdot\left[\nu_k(\boldsymbol{e}_i,\tilde{\mathcal{E}})/(\mu_k(\boldsymbol{e}_i,\mathcal{E})+\epsilon)\right]^{1-\alpha}, \tag{6}$$

where $L(\mathcal{E}) = \tanh\left(\frac{1}{T}\sum_{i=0}^{T}\mu_1(\boldsymbol{e}_i,\mathcal{E})\right)$ is a scaling coefficient.

Here we use the average distance between the states and their nearest neighbors to characterize the intra-episode difference. As a result, the agent lingering in a small region leads to a small L, and the derived intrinsic reward will be decreased. We use the following example to further demonstrate the effectiveness of L. Figure 2 illustrates the trajectories of three episodes. Consider $k = 3, \epsilon = 0.0001$ and $\alpha = 0.5$, the average intrinsic reward computed by Eq. (6) of episode (b) and episode (c) is 0.89 and 0.67, respectively. In episode (b), the agent visited unseen and distinct states when compared to the reference episode, which emphasizes both the intra-episode exploration and exploration across episodes. In contrast, the agent visited novel and repetitive states in episode (c), which will be punished because of low intra-episode difference.

Algorithm 1 REVD

Initialize the encoder $f_{\boldsymbol{\phi}}$ and policy network $\pi_{\boldsymbol{\theta}}$;
Initialize the maximum episodes E, order α, coefficient λ_0, decay rate κ, constant ϵ, and replay buffer $\tilde{\mathcal{E}}, \mathcal{E} \leftarrow \emptyset$;
for $\ell = 1, \ldots, E$ **do**
  Collect the trajectory $\boldsymbol{s}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{a}_{T-1}, \boldsymbol{s}_T$;
  Compute the encoding vectors of $\{\boldsymbol{s}_t\}_{t=0}^{T}$ and $\mathcal{E} \leftarrow \mathcal{E} \cup \{\boldsymbol{e}_t\}_{t=0}^{T}$;
  if $\ell = 1$ **then**
    Set the intrinsic reward as 0;
    $r_t^{\text{total}} \leftarrow \check{r}(\boldsymbol{s}_t, \boldsymbol{a}_t)$;
  else
    Compute the scaling coefficient $L(\mathcal{E})$;
    Compute $\hat{r}(\boldsymbol{s}_t) \leftarrow L(\mathcal{E}) \cdot \left[\nu_k(\boldsymbol{e}_t, \tilde{\mathcal{E}})/\mu_k(\boldsymbol{e}_t, \mathcal{E}) + \epsilon\right]^{1-\alpha}$;
    $r_t^{\text{total}} \leftarrow \check{r}(\boldsymbol{s}_t, \boldsymbol{a}_t) + \lambda_\ell \cdot \hat{r}(\boldsymbol{s}_t)$;
  end if
  Update $\lambda_\ell = \lambda_0(1-\kappa)^\ell$;
  Update the policy network with transitions $\{\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, r_t^{\text{total}}\}$ using any on-policy reinforcement learning algorithm;
  $\tilde{\mathcal{E}} \leftarrow \mathcal{E}, \mathcal{E} \leftarrow \emptyset$.
end for



(a) Reference Episode (b) Good Exploration (c) Limited Exploration

Figure 2: Exploration comparison among three episodes. It is natural to find that Episode (b) achieves better exploration because it visited more distinct states when compared to the reference episode.

4.2 Training Objective

Equipped with the intrinsic reward, the total reward of each transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ is computed as

$$r_t^{\text{total}} = \check{r}(\boldsymbol{s}_t, \boldsymbol{a}_t) + \lambda_t \cdot \hat{r}(\boldsymbol{s}_t), \tag{7}$$

where $\lambda_t = \lambda_0(1-\kappa)^t \geq 0$ is a weighting coefficient that determines the exploration preference, where κ is a decay rate. In particular, this intrinsic reward can be used to perform unsupervised pre-training in the absence of extrinsic rewards $\check{r}$. Then the pre-trained policy can be employed in the downstream task adaptation with extrinsic rewards. Letting $\pi_{\boldsymbol{\theta}}$ denote the policy represented by a neural network with parameters $\boldsymbol{\theta}$, the training objective is to maximize the expected discounted return $\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\sum_{k=0}^{T} \gamma^k r_{t+k+1}^{\text{total}}\right]$. Finally, the detailed workflow of REVD is summarized in Algorithm 1.

5 Experiments

We designed the experiments to answer the following questions:

- Can REVD improve the sample efficiency and policy performance of reinforcement learning algorithms in complex environments with high-dimensional observations? (See Figure 3)
- How does REVD compare to computation-efficient exploration approaches and other exploration approaches empowered by auxiliary models? (See Figure 4 and Figure 8)
- Can REVD achieve remarkable performance in sparse-reward setting? (See Figure 5 and Figure 9)
- Can REVD also improve the sample efficiency in continuous control tasks? (See Figure 7, Figure 8, and Figure 9)

5.1 Atari Games

5.1.1 Setup

We first tested REVD on six Atari games with graphic observations and discrete action space (Brockman et al., 2016). To evaluate the sample efficiency of REVD, two representative reinforcement learning algorithms were introduced to serve as baselines, namely the Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO) (Schulman et al., 2017; Mnih et al., 2016). For comparison with computation-efficient exploration, we selected RE3 (Seo et al., 2021) as the benchmarking scheme, which maximizes the state entropy using a non-parametric entropy estimator. For comparison with exploration approaches empowered by auxiliary models, we selected RIDE (Raileanu & Rocktäschel, 2020) that uses the difference between consecutive states as intrinsic rewards. A brief introduction of RE3 and RIDE can be found in Appendix A. The following results were obtained by setting $k = 5, \alpha = 0.5, \lambda_0 = 0.1, \kappa = 0.00001$ and $\epsilon = 0.0001$, and more details are provided in Appendix B.

5.1.2 Results

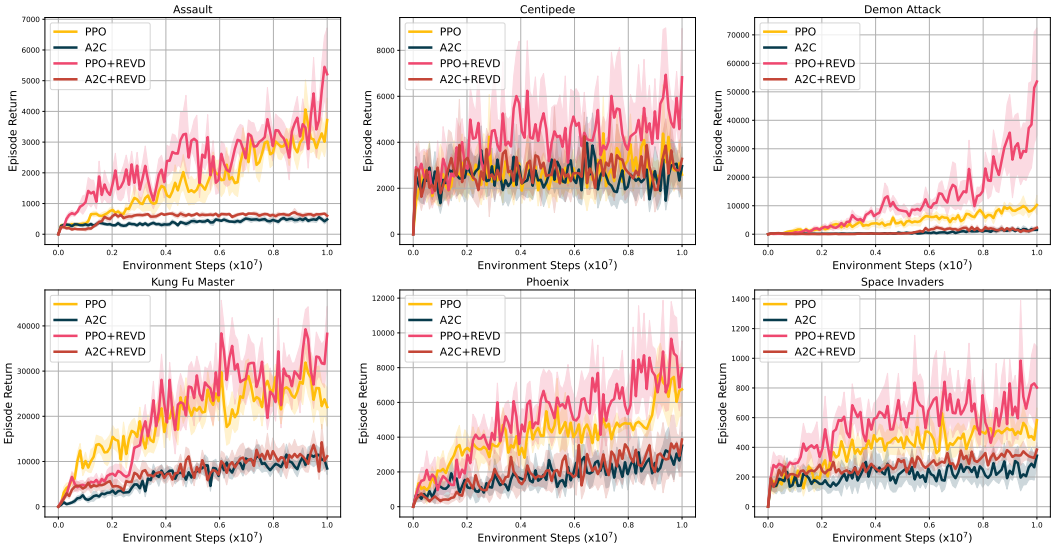


Figure 3: Performance on six Atari games over ten random seeds. REVD significantly promotes the sample efficiency with simple architecture and higher efficiency. The solid line and shaded regions represent the mean and standard deviation, respectively.

Figure 3 illustrates the comparison of the average episode return of six Atari games. It is obvious that REVD significantly improved the sample efficiency of PPO in various tasks. In particular, PPO+REVD achieved an average episode return of 52,000 in *Demon Attack* task, producing a giant performance gain when compared to the vanilla PPO agent. In contrast, the vanilla A2C agent

achieved low performance in these tasks. But REVD also promoted its sample efficiency in most tasks, demonstrating the effectiveness of REVD.

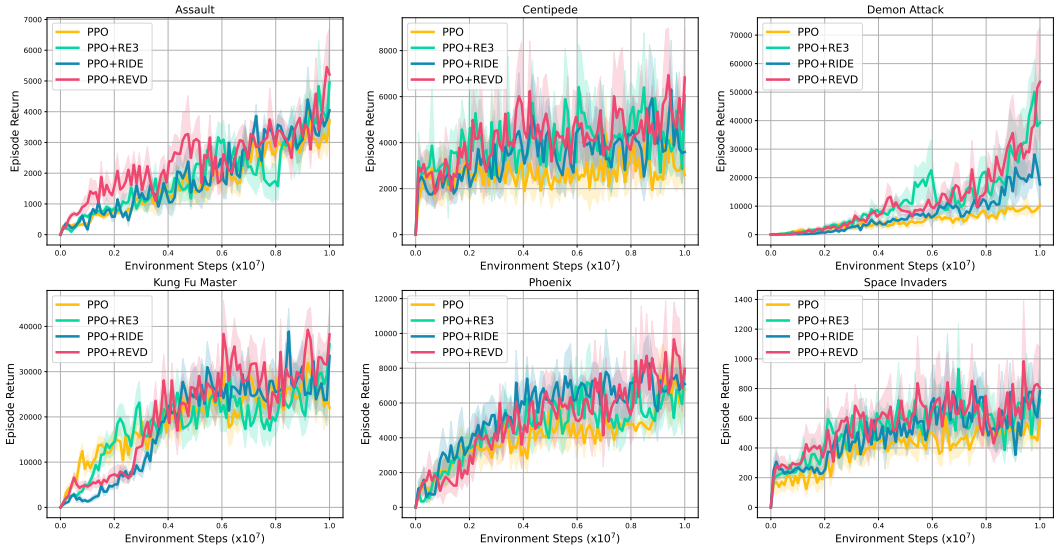


Figure 4: Performance on six Atari games over ten random seeds. PPO+REVD outperforms the benchmarking schemes in all the tasks. The solid line and shaded regions represent the mean and standard deviation, respectively.

Next, we compared REVD with other exploration methods and considered combining them with PPO agent, *i.e.*, PPO+RE3 and PPO+RIDE. As shown in Figure 4, PPO+REVD also achieved the highest policy performance in all tasks. As a computation-efficient method, PPO+RE3 achieved the second highest performance in four tasks. In contrast, PPO+RIDE achieved the second highest performance in two tasks. It is clear that the exploration-empowered agent consistently obtained higher performance than the vanilla PPO agent. Figure 3 and Figure 4 demonstrate that REVD can effectively promote the sample efficiency and policy performance of reinforcement learning algorithms without introducing any auxiliary models and representation learning.

Furthermore, we tested REVD considering the sparse-reward setting, in which the original extrinsic rewards will be randomly set as zero. Figure 5 demonstrates that REVD can still achieve superior sample efficiency in sparse-reward setting. In particular, REVD produced an oscillatory learning curve in *Assault* task, which indicates that REVD provided aggressive exploration incentives. The combination of PPO and RE3 also obtained remarkable performance in all the games, which demonstrates the of high effectiveness and efficiency of computation-efficient exploration.

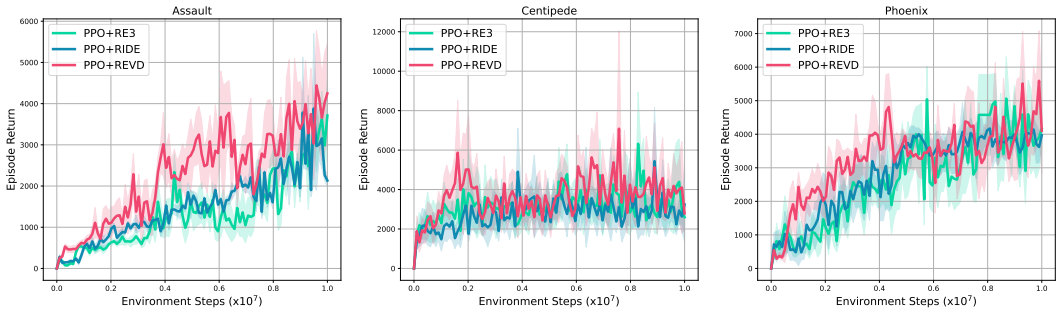
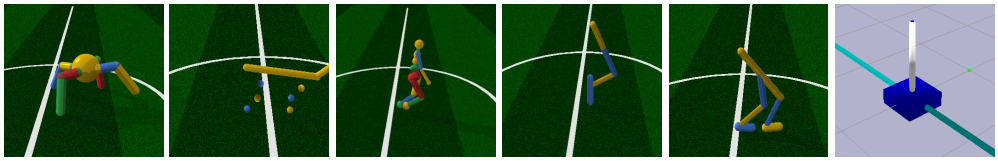


Figure 5: Performance on six Atari games with sparse rewards over ten random seeds. The solid line and shaded area represent the mean and standard deviation, respectively.

5.2 PyBullet Robotics Environments



(a) Ant (b) Half Cheetah (c) Humanoid (d) Hopper (e) Walker 2D (f) Cartpole

Figure 6: Render images of locomotion tasks from PyBullet Robotics Environments. For instance, the goal of *Humanoid* task is to walk forward as fast as possible without falling over.

5.2.1 Setup

Next, we tested REVD using physical simulation tasks of PyBullet Robotics Environments (See Figure 6) Coumans & Bai (2016–2018). We also selected A2C and PPO as the baselines and compared the sample efficiency of REVD with RE3 and RIDE. The following results were obtained by setting $k = 3, \alpha = 0.5, \lambda_0 = 0.1, \kappa = 0.00001$ and $\epsilon = 0.0001$, and more details are provided in Appendix C.

5.3 Results

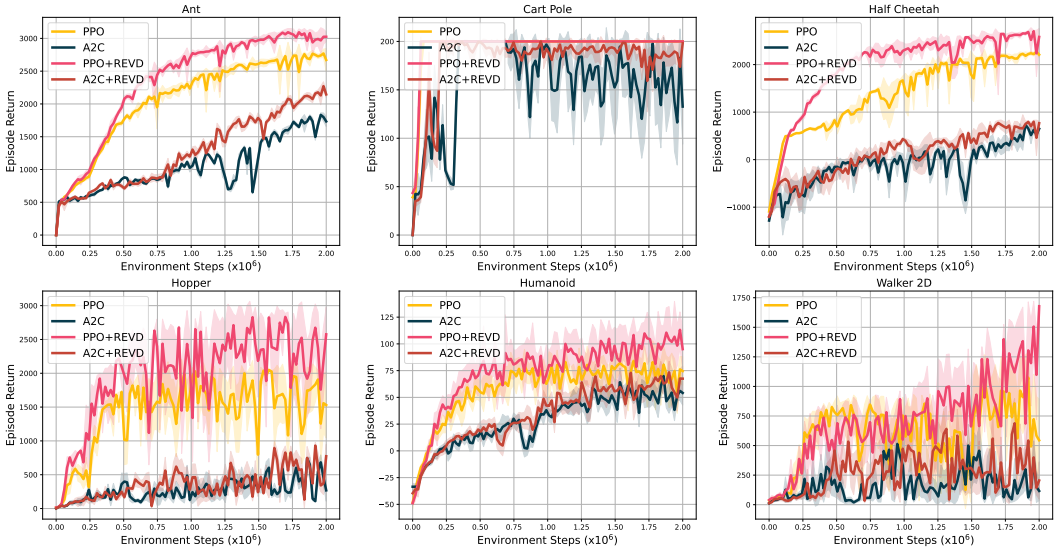


Figure 7: Performance on six PyBullet tasks over ten random seeds. REVD significantly improves the sample efficiency of PPO and A2C. The solid line and shaded regions represent the mean and standard deviation, respectively.

Figure 7 illustrates the comparison of the average episode return of six PyBullet tasks. REVD also significantly promoted the sample efficiency of PPO and A2C in various tasks. In *Walker 2D* task, PPO+REVD obtained the highest average episode return, while the vanilla PPO agent prematurely fell into local optima. In *Cart Pole* task, PPO+REVD solved the problem using the minimum environment steps. For *Hopper* and *Humanoid* task, PPO+REVD achieved the highest convergence rate, demonstrating the powerful exploration capability of REVD. In contrast, A2C achieved low performance in *Half Cheetah* and *Humanoid* task. But A2C+REVD also achieved higher performance than the vanilla A2C agent.

Figure 8 illustrates the comparison of REVD with other exploration methods, and PPO+REVD consistently achieved the highest sample efficiency in most tasks. Finally, we also tested REVD in

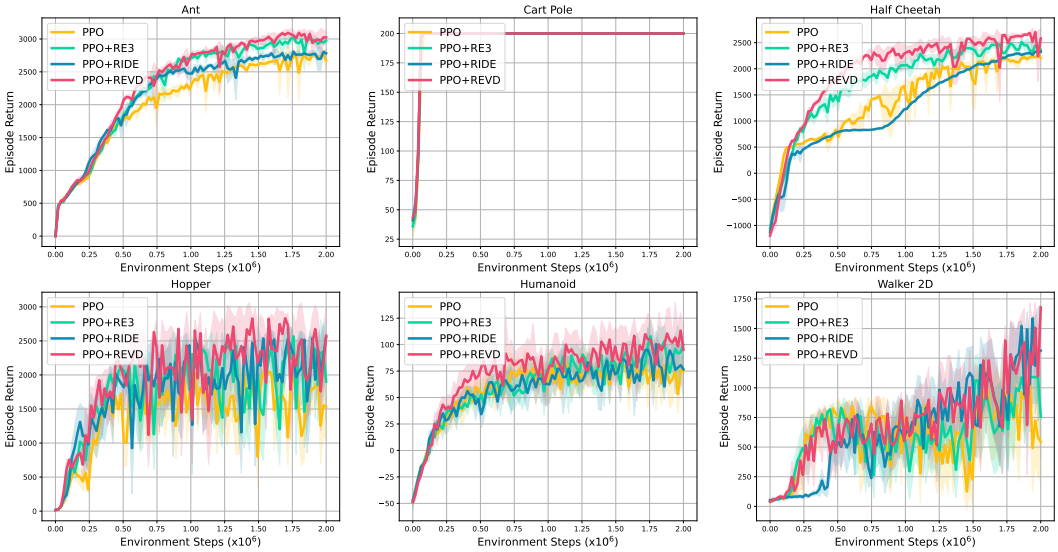


Figure 8: Performance on six PyBullet tasks over ten random seeds. PPO+REVD also outperforms the benchmarking schemes in all the tasks. The solid line and shaded regions represent the mean and standard deviation, respectively.

the sparse-reward setting. Figure 9 demonstrates that REVD also outperformed the benchmarking methods with higher policy performance. Therefore, REVD can effectively improve the sample efficiency in both the continuous and discrete control tasks.

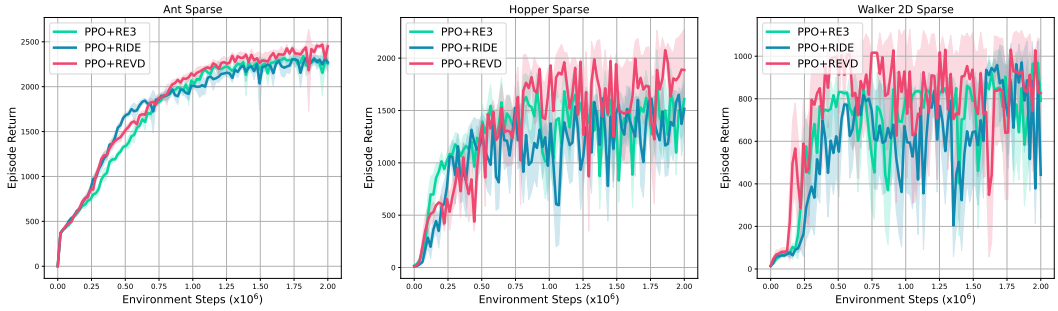


Figure 9: Performance on six PyBullet tasks with sparse rewards over ten random seeds. The solid line and shaded area represent the mean and standard deviation, respectively.

6 Conclusion

In this paper, we propose a novel and computation-efficient exploration method entitled REVD, which provides high-quality intrinsic rewards without auxiliary models or representation learning. Since REVD computes intrinsic rewards based on episodic visitation discrepancy, it can overcome the problem of vanishing rewards and provides sustainable exploration incentives. In particular, REVD provides a quantified metric that reflects the episode-level exploration change. Finally, the REVD is tested on Atari games and PyBullet Robotics Environments. The simulation results demonstrate that REVD and can significantly improve the sample efficiency and outperform the benchmarking schemes. It can be envisaged that the computation-efficient exploration will be widely applied in arbitrary tasks due to its high efficiency and convenience. This work is expected to facilitate and inspire more consequent research on computation-efficient exploration methods.

REFERENCES

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martin Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *Proceedings of the International Conference on Learning Representations*, 2020.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Proceedings of Advances in Neural Information Processing Systems*, 29:1471–1479, 2016.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym. *arXiv preprint arXiv:1606.01540*, 2016.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *Proceedings of the International Conference on Learning Representations*, pp. 1–17, 2019a.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *Proceedings of the 7th International Conference on Learning Representations*, pp. 1–17, 2019b.

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. *URL http://pybullet.org*, 2016–2018.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Proceedings of Advances in neural information processing systems*, 29:1109–1117, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International conference on machine learning*, pp. 448–456, 2015.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. Nonparametric von mises estimators for entropies, divergences and mutual informations. *Advances in Neural Information Processing Systems*, 28, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations*, 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5125–5133, 2020.

Jarryd Martin, Suraj Narayanan S., Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 2471–2478, 2017. doi: 10.24963/ijcai.2017/344. URL https://doi.org/10.24963/ijcai.2017/344.

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.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9028–9036, 2021.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *Advances in neural information processing systems*, 28, 2015.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the International Conference on Machine Learning*, pp. 2721–2730, 2017.

Pierre-Yves Oudeyer and Frederic Kaplan. How can we define intrinsic motivation? In *the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies, Lund: LUCS, Brighton, 2008.

Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Barnabás Póczos and Jeff Schneider. On the estimation of alpha-divergences. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 609–617. JMLR Workshop and Conference Proceedings, 2011.

Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *Proceedings of the International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rkg-TJBFPB`.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *Proceedings of the International Conference on Learning Representations*, 2015.

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

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 9443–9454, 2021.

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

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

Xingrui Yu, Yueming Lyu, and Ivor Tsang. Intrinsic reward driven imitation learning via generative model. In *Proceedings of the International Conference on Machine Learning*, pp. 10925–10935, 2020.

Mingqi Yuan, Man-On Pun, and Dong Wang. Rényi state entropy maximization for exploration acceleration in reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 2022.

Chuheng Zhang, Yuanying Cai, and Longbo Huang Jian Li. Exploration by maximizing Rényi entropy for reward-free RL framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10859–10867, 2021a.

Tianjun Zhang, Paria Rashidinejad, Jiantao Jiao, Yuandong Tian, Joseph E Gonzalez, and Stuart Russell. Made: Exploration via maximizing deviation from explored regions. *Advances in Neural Information Processing Systems*, 34:9663–9680, 2021b.