INHERENT EXPLORATION VIA SAMPLING FOR STOCHASTIC POLICIES

Zhenpeng Shi, Chi Xu, Huaze Tang
Tsinghua Shenzhen International Graduate School, Tsinghua University
{shizp22, xu-c23, tanghz24}@mails.tsinghua.edu.cn

Wenbo Ding*
Tsinghua Shenzhen International Graduate School, Tsinghua University
ding.wenbo@sz.tsinghua.edu.cn

ABSTRACT

In this paper, we propose a novel exploration strategy for reinforcement learning in continuous action spaces by controlling the sampling strategy of stochastic policies. The proposed method, Inherent Exploration via Sampling (IES), enhances exploration by diversifying actions through the selection of varied Gaussian inputs. IES leverages the inherent stochasticity of policies to improve exploration without relying on external bonuses. Furthermore, it integrates seamlessly with existing exploration methods, introducing negligible computational overhead. Theoretically, we prove that IES achieves $\mathcal{O}\left(\epsilon^{-3}\right)$ sample complexity under the actor-critic framework in continuous action spaces. Experimentally, we evaluate IES on Gaussian policies (e.g., Soft Actor-Critic, Proximal Policy Optimization) and consistency-based policies for continuous control benchmarks `mujoco`, `dm_control` and `isaacgym`. The results demonstrate that IES effectively enhances the exploration capabilities of different policies, thereby improving the convergence of various reinforcement learning algorithms.

1 INTRODUCTION

Reinforcement Learning (RL) focuses on learning policies that maximize the expected cumulative reward through interaction with an environment (Sutton, 2018). Unlike Supervised Learning (SL), RL faces two key challenges. First, in RL, data collection is sequential and follows a Markovian process, where future states depend on the current state and action taken, violating the i.i.d. assumption inherent in SL. Second, the objective function—expected cumulative return—is not directly observable and therefore requires careful estimation and approximation.

Exploration (Ladosz et al., 2022; Hao et al., 2023) plays a pivotal role in addressing key challenges in Reinforcement Learning (RL). Effective exploration enables the agent to gather high-quality data, which is essential for both policy evaluation and policy improvement. By obtaining better data, the agent can achieve more accurate estimations of the RL objective, thereby enhancing the performance of RL algorithms.

Current exploration strategies can be broadly classified into three categories. **Noise Injection**. Exploration in continuous action spaces can be achieved by perturbing actions with state-dependent noise (Rückstieß et al., 2008), or by introducing temporally correlated noise (Lillicrap, 2015). State-dependent noise injection (Rückstieß et al., 2008) modulates exploration magnitude based on state uncertainty estimates. Temporally correlated noise via Ornstein-Uhlenbeck processes (Lillicrap, 2015) enables structured exploration in physical control tasks. Another approach is injecting noise into the parameter space(Plappert et al., 2017), which perturbs policy network weights to induce diverse behavior while maintaining action smoothness. **Exploration Bonus**. An exploration bonus can be added to the reward function to promote exploration. Maximum entropy regularization

*Corresponding author

(Haarnoja et al., 2018) encourages diverse actions by maximizing policy entropy. Information-theoretic bonuses (Houthooft et al., 2016) quantify epistemic uncertainty through variational inference. Pseudocount-based methods (Tang et al., 2017) estimate state novelty using density models. Flip exploration (Lobel et al., 2023) induces diverse trajectories through action sequence inversion. Curiosity-driven exploration (Pathak et al., 2017) rewards prediction errors in learned dynamics models. Wasserstein optimistic exploration (Likmeta et al., 2023) constructs confidence bounds in Wasserstein space for safe exploration. **Policy Distribution**. Several works have explored more complex policy distributions to enhance exploration. These include normalizing flows (Ward et al., 2019), diffusion models (Ren et al., 2024), and consistency models (Ding & Jin, 2023; Chen et al., 2023). There are some other approaches that are not exactly categorized by these categories. For example, Random Latent Exploration (RLE) (Mahankali et al., 2024) conditions the policy by random goal vector to encourage exploration.

However, we argue that most existing approaches do not fully exploit the inherent stochasticity of the policies. In this work, we focus on developing a novel exploration strategy that leverages the full potential of this inherent stochasticity by controlling the sampling strategy. The contributions of this work are threefold:

1. We propose a novel exploration strategy, *Inherent Exploration via Sampling (IES)*, specifically designed for reinforcement learning in continuous action spaces. This method effectively controls the sampling behavior of stochastic policies to enhance exploration efficiency.
2. We provide a theoretical analysis of IES and prove that it achieves a sample complexity of $\mathcal{O}(\epsilon^{-3})$.
3. We empirically evaluate IES on continuous control tasks across three distinct reinforcement learning algorithms, comparing its performance with other state-of-the-art exploration strategies. The results show that IES enhances exploration efficiency and can be seamlessly integrated with existing exploration techniques to achieve further improvements.

2 PRELIMINARY

The Markov decision process in reinforcement learning can be formalized as a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, R, \gamma, \rho_0)$. Where $\mathcal{S}$ represents a bounded d_s dimensional continuous state space, $\mathcal{A}$ represents a bounded d_a dimensional continuous action space. $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(S)$ represents the probability transition function, where $\mathcal{P}(S)$ represents a set of distribution on the state space $\mathcal{S}$. $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$ represented the reward function. $\gamma \in (0, 1)$ represents the discount factor, and ρ_0 represent a initial distribution over state space.

The policy $\pi : \mathcal{S} \to \mathcal{P}(A)$ maps a state to a distribution over the action space. In the context of continuous control, the policy can often be represented by a sampling function: $a = f(s, z)$, where $z \sim \mathcal{N}(0, I_{d_a})$ is a d_a-dimensional sample from a standard Gaussian distribution. In deep reinforcement learning, this sampling function $a = f_\phi(s, z)$ is parameterized by a neural network with parameters $\phi \in \Phi$, where Φ represents the entire parameter space of the policy network. We denote the policy corresponding to the sampling function $a = f_\phi(s, z)$ as $\pi_\phi : \mathcal{S} \to \mathcal{P}(A)$. This general formulation encompasses a variety of policy types, including Gaussian policies (Haarnoja et al., 2018), diffusion policies (Chi et al., 2023), and consistency policies (Chen et al., 2023), among others.

The state action value function $Q^\pi(s, a)$ denotes the expected cumulative reward starting from state s_0 and action a_0 and following policy π.

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right], \tag{1}$$

where the expectation is taken with respect to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$ and $a_t \sim \pi(\cdot \mid s_t)$.

In the context of actor-critic algorithms, the goal of reinforcement learning is to maximize the expected cumulative return

$$J(\phi) = \mathbb{E}_{s_0 \sim \rho_0, a \sim \pi_\phi(\cdot|s)}\left[Q^{\pi_\phi}(s, a)\right], \tag{2}$$

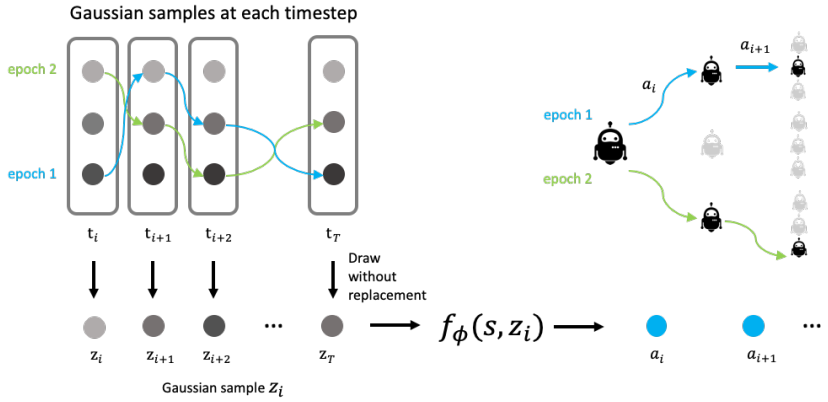


Figure 1: Illustration of the Inherent Exploration via Sampling (IES) method. First, we generate Gaussian samples for each timestep using the quasi-Monte Carlo method. Secondly, at each timestep, we randomly select one Gaussian sample z *without replacement*, and use the selected Gaussian sample to compute the action via $a = f_\phi(s, z)$.

where ρ_0 denotes the initial state distribution.

The discounted state visitation measure is defined as

$$d_{\rho_0}^{\pi_\phi}(\cdot) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}_{\rho_0,\pi_\phi}(s_t \in \cdot) \tag{3}$$

where $\gamma \in [0, 1)$ is the discount factor, and $\mathbb{P}_{\rho_0,\pi_\phi}(s_t \in \cdot)$ represents the probability of the state s_t being in the set $\cdot$ under the state distribution ρ_0 and the policy π_ϕ. And let $\zeta_{\rho_0}^{\pi_\phi}(\cdot)$ denote the stationary state-action distribution starts from initial state distribution ρ_0 and follows policy π_ϕ.

3 Inherent Exploration via Sampling

Inherent Exploration via Sampling (IES) encourages exploration in online reinforcement learning environments by directly controlling the policy's sampling behavior. Specifically, it selects the Gaussian input z_i at timestep i before passing it to the policy sampling function, $a_i = f(s_i, z_i)$. By carefully selecting the Gaussian input to be both diverse and representative, IES ensures that the actions are diverse while maintaining the unbiasedness of policy evaluation and policy improvement.

The main process of Inherent Exploration via Sampling (IES) consists of two steps. First, we generate n^e diverse and representative Gaussian samples, denoted as $\{z_i^k\}_{k=1}^{n_e}$, for each timestep i. Second, at each timestep i during an episode, we select one Gaussian sample, z_i, from the set $\{z_i^k\}_{k=1}^{n_e}$ without replacement and use the policy sampling function $a_i = f(s_i, z_i)$ to compute the action. This action is then used in the online interaction. When all Gaussian samples at timestep i are exhausted, new samples $\{z_i^k\}_{k=1}^{n_e}$ are generated.

3.1 Generate Gaussian Samples via Quasi-Monte Carlo

The first step of IES is to generate sufficiently representative Gaussian samples. Those samples will be stored and used in online interactions later.

In this work, we utilize randomized quasi-Monte Carlo (QMC) methods (Niederreiter, 1992; Owen, 2004; 2008) to generate evenly distributed Gaussian samples. QMC methods, commonly used for numerical integration, rely on low-discrepancy sequences such as the Faure, Sobol', and Halton sequences (see (Caflisch et al., 1997; Niederreiter, 1992; Owen, 2013)) to achieve a faster convergence rate of $O(n^{-1+\epsilon})$, where $\epsilon > 0$ is an arbitrarily small constant. This enhanced convergence rate is derived from the Koksma-Hlawka inequality (Hlawka & Mück, 1972), which relates the integration error to two factors: the variation of the integrand (measured in the Hardy-Krause sense) and the uniformity of the sample points, characterized by their star discrepancy.

As shown in Figure 2, quasi-Monte Carlo (QMC) samples form a more structured and uniform pattern than Monte Carlo (MC) samples for a 2D Gaussian distribution. This structure enables more efficient action diversification by better leveraging policy stochasticity without external guidance.

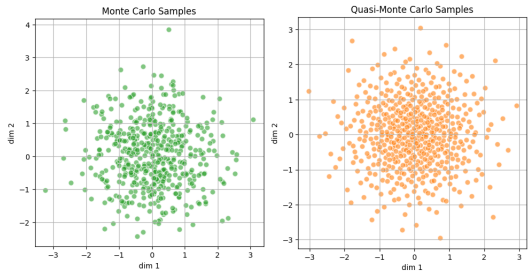


Figure 2: Monte Carlo and Quasi-Monte Carlo Sampling.

3.2 Control Samples in online interaction

In online interaction at timestep i, IES randomly picks one Gaussian sample z_i from the pre-generated Gaussian samples$\{z_i\}_{i=0}^{n^e}$ without replacement. The selected Gaussian sample is then passed to the policy function to get action $a_i = f_\phi(s_i, z_i)$. Then the action a_i is used in the interaction with the environment.

Algorithm 1 Inherent Exploration via Sampling (IES)

Initialize Gaussian samples $\{z_i\}_{i=0}^{n^e}$ via randomized quasi-Monte Carlo method.
for each iteration **do**
  for each environment step t **do**
    if $\{z_i\}_{i=0}^{n^e}$ are all selected **then**
      Generate new Gaussian samples $\{z_i\}_{i=0}^{n^e}$.
    end if
    Randomly select a Gaussian input z_t from the generated Gaussian samples $\{z_i\}_{i=0}^{n^e}$ without replacement.
    Compute action $a_t = f(s_t, z_t)$.
    Observe s_{t+1}, r_t from the environment after executing a_t.
    Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer.
  end for
  for each gradient step **do**
    Critic update and policy update.
  end for
end for

The no-replacement strategy encourages the agent to explore new actions while still keeping the objective function unchanged. After the $\{z_i\}_{i=0}^{n^e}$ samples are all used, there is a new round of generation of $\{z_i\}_{i=0}^{n^e}$ quasi-Monte Carlo samples.

It is worth noticing that IES only affects the sampling in the action computation. Thus it can be applied to many existing exploration methods such as noise injection (Lillicrap, 2015) and bonus (Haarnoja et al., 2018), etc.

4 Theoretical Analysis

In this section, we analyze the behavior of Inherent Exploration via Sampling and provide a global convergence guarantee under common assumptions. The analysis is highly inspired by the framework of actor-critic algorithms under neural network approximation (Gaur et al., 2024).

Policy Regularity (Liu et al., 2020)

Assumption 4.1. Let $\phi, \phi_1, \phi_2 \in \Phi$ and $(s, a) \in \mathcal{S} \times \mathcal{A}$. The following hold:

$$\|\nabla_\phi \log(\pi_{\phi_1}(a \mid s)) - \nabla_\phi \log(\pi_{\phi_2}(a \mid s))\| \leq l_1 \|\phi_1 - \phi_2\|,$$
$$\|\nabla_\phi \log(\pi_\phi(a \mid s))\| \leq M_g,$$

where $l_1 > 0$ and $M_g > 0$ are constants.

This assumption is a standard assumption regarding policy regularity, and it has been widely used in prior works such as (Liu et al., 2020; Fatkhullin et al., 2023; Gaur et al., 2024). It assumes that the gradient of the log-policy satisfies the Lipschitz continuity property and the norm of the gradient of the log-policy is bounded. Gaussian policies are proven to fulfill this assumption (Liu et al., 2020).

Fisher-non-degenerate Policy (Zhang et al., 2020)

Assumption 4.2. Let $(s, a) \in (\mathcal{S} \times \mathcal{A})$ and $\phi \in \Phi$. The following conditions hold:

$$\mathbb{E}_{(s,a)\sim d_{\rho_0}^{\pi_{\phi_1}}} \left(\nabla_\phi \log \pi_{\phi_1}(a \mid s) \cdot \nabla_\phi \log \pi_{\phi_1}(a \mid s)^\top \right)$$
$$\succeq \mu_f I_d,$$

where $\mu_f \geq 0$ and I_d is the identity matrix of dimension d.

This is a common assumption in the literature (Fatkhullin et al., 2023). It is shown to be satisfied by Gaussian policies under specific parameterization(Fatkhullin et al., 2023).

Ergodicity (Xu et al., 2020)

Assumption 4.3. Let $\phi \in \Phi$ and π_ϕ be the corresponding policy. We assume the following: There exists a positive integer p such that for every positive integer k, for any measurable set $\cdot \in \mathcal{S} \times \mathcal{A}$, and for any initial state $s \in \mathcal{S}$, the total variation distance satisfies:

$$d_{TV} \left(\mathbb{P}\big((s_k, a_k) \in \cdot \mid s_0 = s\big), \zeta_{\rho_0}^{\pi_\phi}(\cdot) \right) \leq p\rho^k,$$

where $\rho \in [0, 1)$ and d_{TV} denotes the total variation distance.

This ergodicity assumption is standard in reinforcement learning under Markovian sampling (Bhandari et al., 2018; Xu et al., 2020; Gaur et al., 2024).

Compatible Policy Representation (Fatkhullin et al., 2023)

Assumption 4.4. There exists $\epsilon_{bias} \geq 0$ such that for every $\phi \in \Phi$, the estimation error satisfies:

$$\mathbb{E}\left[\left(A^{\pi_\phi}(s, a) - \left((1-\gamma) w^*(\phi)^\top \nabla_\phi \log \pi_\phi(a \mid s) \right) \right)^2 \right]$$
$$\leq \epsilon_{bias},$$

where $Q^{\pi_\phi}(s, a)$ is the state-action value function, and $w^*(\phi)$ is defined as:

$$w^*(\phi) := F_\rho(\phi)^\dagger \nabla_\phi J(\phi),$$

where $F_{\rho_0}(\phi)^\dagger$ is the pseudo-inverse of the Fisher information matrix $F_{\rho_0}(\phi)$. The expectation is taken over $s \sim d_{\rho_0}^{\pi^*}$ and $a \sim \pi^*(\cdot \mid s)$, where π^* is the optimal policy that maximizes $J(\phi)$.

This assumption is widely used in reinforcement learning works with last iterate convergence analysis (Fatkhullin et al., 2023; Gaur et al., 2024)

Critic Approximation (Gaur et al., 2024)

Assumption 4.5. For any fixed $\phi \in \Phi$, we have:

$$\min_{\theta \in \Theta'} \mathbb{E}_{s,a \sim \zeta_{\rho_0}^{\pi_\phi}} \left(Q_\theta(s,a) - Q^{\pi_\phi}(s,a)\right)^2 \leq \epsilon_{\text{approx}},$$

where T^{π_ϕ} is the Bellman operator associated with policy π_ϕ, $\zeta_{\rho_0}^{\pi_\phi}$ is the stationary state-action distribution, and ϵ_{approx} represents the approximation error.

4.1 Main Theorem

Theorem 1 (Global Convergence of IES)

Suppose Assumptions 4.1, 4.2, 4.3, 4.4, and 4.5 hold. Then we have:

$$\begin{aligned} J^* - J(\phi_T) &\leq \mathcal{O}\left(\frac{1}{T}\right) \\ &\quad + \frac{1}{T^2} \sum_{t=1}^{T-1} (t+1) \frac{16}{\sqrt{\mu'}} \left[\|\nabla J(\phi_t) - d_t\|\right], \\ &\leq \mathcal{O}\left(\frac{1}{T}\right) + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right), \end{aligned}$$

Where the J^* denotes the optimal expected return and $J(\phi_T)$ denotes the expected return for π_{ϕ_T}, n denotes the number of samples collected by *IES* method in one gradient update, $\mu' = \frac{\mu_F^2}{2M_g^2}$, $\nabla J(\phi_t)$ denotes the ground truth gradient, d_t denotes the estimated gradient, and $\tilde{O}(\cdot)$ represents asymptotic complexity. Thus the sample complexity is $\mathcal{O}(\epsilon^{-3})$.

5 Experimental Results

In this section, we experiment with Inherent Exploration via Sampling (IES) for continuous control. First, we demonstrate that the stochasticity in policies is an abundant resource that can be leveraged for exploration. Second, we show that IES can significantly enhance exploration compared to other baseline methods. Third, we provide an extensive evaluation of IES across three major continuous control environments: `mujoco`, `dm_control`, and `isaacgym`. IES is tested on several reinforcement learning algorithms, including Proximal Policy Optimization (PPO) (Schulman et al., 2017), Soft Actor-Critic (SAC) (Haarnoja et al., 2018), and Consistency Policy Q Learning (CPQL) (Chen et al., 2023). Finally, we offer additional discussions regarding the proposed methods. The implementation of the proposed method will be made publicly available upon acceptance.

Table 1: Performance comparison of different methods across `MuJoCo` v4 environments.

Method	Swimmer	HalfCheetah	Ant	Humanoid	Humanoid-Standup
PPO	110	1497	850	632	148889
SAC	48	11334	1821	5194	157336
CPQL	113	9491	3598	4911	136646
PPO + IES (Ours)	**116**	1524	1398	520	138246
SAC + IES (Ours)	56	**11407**	**5011**	**5621**	**159396**
CPQL + IES (Ours)	106	9807	4048	5268	122673

5.1 Stochasticity in Policies

First, we show that policies can contain great stochasticity. The following Figure 3 shows the action distribution learned by consistency policy q learning (CPQL) (Chen et al., 2023). We trained CPQL

Table 2: Performance comparison across `dm_control` environments. Abbreviations: A-S (Acrobot-Swingup), C-S (Cartpole-Swingup), F-S (Finger-Spin), F-TE (Finger-TurnEasy), F-TH (Finger-TurnHard), H-H (Hopper-Hop), W-R (Walker-Run).

Method	A-S	C-S	Cheetah	F-S	F-TE	F-TH	H-H	W-R
PPO	34.4	760.7	560.4	369.9	275.2	5.0	2.2	131.7
SAC	33.2	781.4	530.9	825.4	371.4	344.8	270.9	445.9
PPO + IES (Ours)	**80.2**	744.7	340.6	646.0	219.0	93.6	27.2	100.6
SAC + IES (Ours)	18.0	**867.3**	**796.5**	**969.3**	**727.2**	**990.0**	**607.9**	**672.5**

in `mujoco` Swimmer environment for 1 million steps. Then we randomly select a state s in the state space, generate n^e quasi-Monte Carlo samples (Figure 3 left), and pass them to the policy function to get n^e action samples (Figure 3 right). The actions are transformed by the PCA method for visualization purposes. It can be shown that the stochastic policy can model complex non-Gaussian distribution in high-dimensional action space.

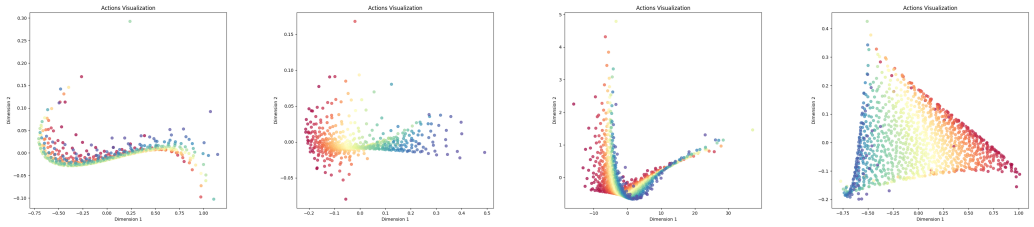
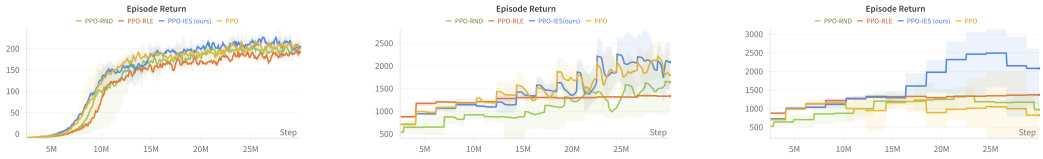
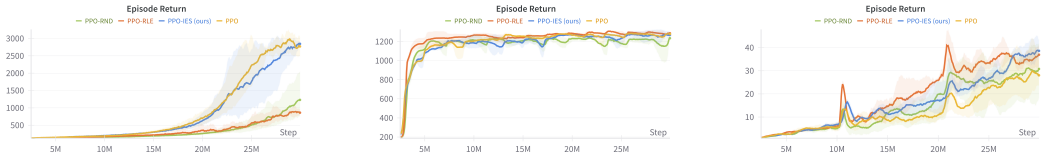


Figure 3: Actions sampled from stochastic policies learned by CPQL (Chen et al., 2023) in the `mujoco` Swimmer environment. Each subfigure represents action samples for a random state in the Swimmer environment.



(a) AllegroHand (b) FrankaCabinet (c) FrankaCabinetTimeOut



(d) Humanoid (e) Quadcopter (f) Anymal

Figure 4: Performance visualization on IsaacGym environments: (a) AllegroHand, (b) FrankaCabinet, (c) FrankaCabinetTimeOut, (d) Humanoid, (e) Quadcopter, and (f) Placeholder.

5.2 Experiments Results on Online Environments

In this section, we show the experiment results for IES in online reinforcement learning tasks in continuous control environments.

Environments. We evaluate IES on 3 major environments for continuous control: `mujoco` (Todorov et al., 2012), `dm_control` (Tunyasuvunakool et al., 2020), and

`IssacGym` (Makoviychuk et al., 2021). These environments provide diverse and complex benchmarks for robotic locomotion and manipulation.

Backbone Algorithms. We consider 3 reinforcement learning algorithms as backbone algorithms for exploration: Soft Actor-Critic (SAC) (Haarnoja et al., 2018) and Proximal Policy Optimization (Schulman et al., 2017) are using Gaussian policies, and Consistency Policy Q Learning (CPQL) (Chen et al., 2023) is using simplified consistency policies. The backbone implementation of SAC and PPO is based on CleanRL (Huang et al., 2022) project and the implementation of CPQL follows the CPQL original paper (Chen et al., 2023).

Exploration Baselines. Exploration baselines include Random Network Distillation (Burda et al., 2018), and Random Latent Exploration (Mahankali et al., 2024).

Does *IES* improve the performance of Backbone algorithms? First, we compare the performance of the backbone algorithms (PPO, SAC, CPQL) with the backbone algorithms enhanced by IES (PPO-IES, SAC-IES, CPQL-IES). Each algorithm is trained for 1 million steps across 3 random seeds. Evaluation is performed every 10,000 steps, and we report the evaluation return curve during training in Figure 6. The line represents the mean and the shaded area represents the minimum and maximum returns across different seeds.

Table 1 reports the evaluation mean after training for 1 million steps in `MuJoCo`. The results indicate that IES improves the performance of most backbone algorithms across various environments. Table 2 provides the evaluation mean for `dm_control`, where each IES algorithm is trained for 50k steps using 3 random seeds, and the results of PPO and SAC are taken from the CPQL paper (Chen et al., 2023). SAC + IES consistently achieves the highest returns in complex tasks such as Finger-TurnHard and Walker-Run, highlighting its superior exploration capabilities. PPO + IES shows notable improvements in environments like Acrobot-Swingup. However, it is worth noting that the performance improvement of IES for SAC is relatively larger than for PPO. While PPO generally benefits from IES, there are occasional cases where its performance slightly decreases.

How Does *IES* Compare to Other Exploration Algorithms? Secondly, we compare the performance of IES with other exploration strategies in IsaacGym environments. The results in Table 3 and the training curve in Figure 4 highlight the effectiveness of PPO-IES compared to other methods. PPO-IES achieves the highest returns in challenging environments such as FrankaCabinet, Humanoid, and FrankaCabinetTimeout, demonstrating its ability to enhance exploration and improve policy performance. While PPO-RND performs well in ShadowHand, and PPO achieves the best result in BallBalance, PPO-IES consistently outperforms in more complex and diverse tasks, validating its robustness and adaptability across different IsaacGym environments.

6 Discussion

Computational Cost The only additional computational cost is the generation of quasi-Monte Carlo Gaussian samples. The following action computation, online interaction, and learning follow the same procedure as regular reinforcement learning algorithms. The additional computational cost is negligible. **Applicability** The IES method only affects the action computation step. It can be applied to any policy class in the form of sampling function $a = f(s, z)$. These include Gaussian policies, and many generative models. Furthermore, it can be easily applied to existing online exploration methods like noise injection or bonus exploration. **Limitations and Future Directions**. In this work, we have focused on sampling, future research could explore other active sampling strategies to further enhance exploration. Additionally, our current experiments are limited to simulation environments, and future work could involve testing the proposed methods on real robotic platforms to validate their performance in practical scenarios.

7 Conclusion

We introduced *Inherent Exploration via Sampling (IES)*, a novel exploration strategy tailored for reinforcement learning in continuous action spaces. Our theoretical analysis demonstrates that IES achieves a sample complexity of $\mathcal{O}(\epsilon^{-3})$. Empirical evaluations across multiple algorithms and tasks confirm that IES enhances exploration efficiency and integrates seamlessly with existing methods to achieve superior performance.

REFERENCES

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pp. 1691–1692. PMLR, 2018.

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

Russel E. Caflisch, William J. Morokoff, and Art B. Owen. Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. *Journal of Computational Finance*, 1: 27–46, 1997.

Yuhui Chen, Haoran Li, and Dongbin Zhao. Boosting continuous control with consistency policy. *arXiv preprint arXiv:2310.06343*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Yuhao Ding, Junzi Zhang, and Javad Lavaei. On the global optimum convergence of momentum-based policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pp. 1910–1934. PMLR, 2022.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. *arXiv preprint arXiv:2309.16984*, 2023.

Ilyas Fatkhullin, Anas Barakat, Anastasia Kireeva, and Niao He. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. In *International Conference on Machine Learning*, pp. 9827–9869. PMLR, 2023.

Mudit Gaur, Amrit Singh Bedi, Di Wang, and Vaneet Aggarwal. Closing the gap: Achieving global convergence (last iterate) of actor-critic under markovian sampling with neural network parametrization. *arXiv preprint arXiv:2405.01843*, 2024.

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

Jianye Hao, Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Zhaopeng Meng, Peng Liu, and Zhen Wang. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Edmund Hlawka and R Mück. Über eine transformation von gleichverteilten folgen II. *Computing*, 9:127–138, 1972.

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

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and JoÃĢo GM AraÃšjo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

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

Amarildo Likmeta, Matteo Sacco, Alberto Maria Metelli, and Marcello Restelli. Wasserstein actor-critic: directed exploration via optimism for continuous-actions control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8782–8790, 2023.

TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33:7624–7636, 2020.

Sam Lobel, Akhil Bagaria, and George Konidaris. Flipping coins to estimate pseudocounts for exploration in reinforcement learning. In *International Conference on Machine Learning*, pp. 22594–22613. PMLR, 2023.

Srinath Mahankali, Zhang-Wei Hong, Ayush Sekhari, Alexander Rakhlin, and Pulkit Agrawal. Random latent exploration for deep reinforcement learning. *arXiv preprint arXiv:2407.13755*, 2024.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, PA, 1992.

Art B Owen. Quasi-Monte Carlo for integrands with point singularities at unknown locations. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pp. 403–417. Springer, 2004.

Art B. Owen. Local antithetic sampling with scrambled nets. *The Annals of Statistics*, 36(5):2319–2343, 2008.

Art B Owen. *Monte Carlo Theory, Methods and Examples*. Stanford, 2013.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

Thomas Rückstieß, Martin Felder, and Jürgen Schmidhuber. State-dependent exploration for policy gradient methods. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part II 19*, pp. 234–249. Springer, 2008.

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

Sebastian U Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.

Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

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.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

Patrick Nadeem Ward, Ariella Smofsky, and Avishek Joey Bose. Improving exploration in soft-actor-critic with normalizing flows policies. *arXiv preprint arXiv:1906.02771*, 2019.

Tengyu Xu, Zhe Wang, and Yingbin Liang. Improving sample complexity bounds for (natural) actor-critic algorithms. *Advances in Neural Information Processing Systems*, 33:4358–4369, 2020.

Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Basar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6): 3586–3612, 2020.

A Related Works

A.1 Stochastic Policies

In the context of continuous action space reinforcement learning, policies are typically represented as stochastic distributions. Common examples of stochastic policies include Gaussian policies (Haarnoja et al., 2018; Schulman et al., 2017), Diffusion Policies (Chi et al., 2023), and Consistency Policies (CP) (Ding & Jin, 2023). Gaussian policies, widely adopted in algorithms like Soft Actor-Critic (Haarnoja et al., 2018) and Proximal Policy Optimization (Schulman et al., 2017), model actions through mean and variance parameters. Diffusion Policies (Chi et al., 2023) utilize iterative denoising processes inspired by generative models to generate temporally consistent actions. Consistency Policies (CP) (Ding & Jin, 2023) employ consistency models that enable single-step policy evaluation while maintaining multi-step training benefits within actor-critic architectures. Consistency Policy Q Learning (CPQL) (Chen et al., 2023) simplifies CP training through a modified Q-learning objective that bypasses complex consistency constraints.

A.2 Quasi-Monte Carlo

Efficient numerical integration is crucial in statistics, finance, and reinforcement learning, where computing expectations is common. Given an integral $\mathbb{E}[f(Z)]$ with $Z \sim \mathcal{U}[0,1]^d$, the standard Monte Carlo (MC) estimator is:

$$\widehat{I}_n = \frac{1}{n}\sum_{i=1}^{n} f(y_i),$$

where $\{y_i\}_{i=1}^n$ are i.i.d. samples from $\mathcal{U}[0,1]^d$. The MC method achieves a convergence rate of $\mathcal{O}(n^{-1/2})$ due to the central limit theorem.

In contrast, Quasi-Monte Carlo (QMC) methods replace random sampling with low-discrepancy sequences, such as Sobol', Faure, and Halton sequences, to improve integration accuracy. The error bound follows the Koksma-Hlawka inequality (Hlawka & Mück, 1972):

$$\left| \int f(y)dy - \frac{1}{n}\sum_{i=1}^{n} f(y_i) \right| \leq V_{\text{HK}}(f) D_n^*,$$

where $V_{\text{HK}}(f)$ measures function variation, and D_n^* denotes the star discrepancy, quantifying the uniformity of sample distribution. QMC sequences reduce discrepancy to $\mathcal{O}(n^{-1}(\log n)^d)$, outperforming MC in convergence speed when $V_{\text{HK}}(f)$ is finite.

Randomized Quasi-Monte Carlo (RQMC) further enhances QMC by introducing controlled randomness while maintaining low discrepancy. Methods such as scrambling and random shifting ensure that points remain uniformly distributed while preserving deterministic structure. RQMC retains a convergence rate of $\mathcal{O}(n^{-1+\epsilon})$ and, under smooth conditions, can achieve $\mathcal{O}(n^{-3/2+\epsilon})$ (Owen, 2008).

B Theory

B.1 Lemmas

Lemma B.1 (Relaxed weak gradient domination, (Ding et al., 2022))**.** *Let Assumptions 1-(ii), 2, and 4 hold. Then*

$$\forall \phi \in \mathbb{R}^d, \quad \epsilon' + \|\nabla J(\phi)\| \geq \sqrt{2\mu'}\left(J^* - J(\phi)\right), \tag{4}$$

where

$$\epsilon' = \frac{\mu_F \sqrt{\epsilon_{bias}}}{M_g(1-\gamma)} \quad \textit{and} \quad \mu' = \frac{\mu_F^2}{2M_g^2}. \tag{5}$$

Lemma B.2 (Relaxed Descent Lemma for Policy Gradient, (Fatkhullin et al., 2023; Gaur et al., 2024))**.** *Let Assumptions 1, 2, 3, and 4 hold. Let $J(\phi_t)$ denote the expected return for the policy*

parameterized by ϕ_t at iteration t. Under the smoothness property of the expected return $J(\phi)$, the following holds:

$$-J(\phi_{t+1}) \leq -J(\phi_t) - \frac{\alpha_t}{3}\|\nabla J(\phi_t)\| + \frac{8\alpha_t}{3}\|e_t\| + L_J\|\phi_{t+1} - \phi_t\|^2, \tag{6}$$

where α_t is the step size at iteration t, e_t is the error term defined as $e_t = d_t - \nabla J(\phi_t)$, and L_J is the smoothness constant of $J(\phi)$.

Lemma B.3 (Recursion Lemma, (Fatkhullin et al., 2023) Lemma 12, (Stich, 2019) Lemma 7.)**.** *Let a be a positive real number, τ a positive integer, and let $\{r_t\}_{t\geq 0}$ be a non-negative sequence satisfying for every integer $t \geq 0$:*

$$r_{t+1} - r_t \leq -a\alpha_t r_t + \beta_t, \tag{7}$$

where $\{\alpha_t\}_{t\geq 0}$ and $\{\beta_t\}_{t\geq 0}$ are non-negative sequences and $a\alpha_t \leq 1$ for all t. Then, for $\alpha_t = \frac{2}{a(t+\tau)}$, we have for every integer $t_0, T \geq 1$:

$$r_T \leq \frac{(t_0 + \tau - 1)^2 r_{t_0}}{(T + \tau - 1)^2} + \frac{\sum_{t=t_0}^{T-1} \beta_t (t+\tau)^2}{(T+\tau-1)^2}. \tag{8}$$

Lemma B.4 (Bounded Gradient Error Lemma)**.** *Let d_t be the estimated gradient at gradient step t, and let n denote the number of state action pairs used in the gradient estimation. Suppose that all the state action pairs are collected by policy π_{ϕ_t} under the IES method. Then we have,*

$$\|\nabla J(\phi_t) - d_t\| = \|\nabla J(\phi_t) - \frac{1}{n}\sum_{i=1}^{n} \nabla log\pi_{\phi_t}(a_i|s_i) Q_{\theta_t}(a_i, s_i)\| \tag{9}$$

$$\leq \mathcal{O}(\frac{1}{\sqrt{n}}) \tag{10}$$

B.2 Proofs

Proof of theorem 1.

Proof. Under Assumptions 4.1, 4.2, 4.3, and 4.4, plugging the inequality of Lemma B.1 into the inequality of Lemma B.2, we get:

$$\begin{aligned}
-J(\phi_{t+1}) &\leq -J(\phi_t) - \frac{\alpha_t\sqrt{\mu'}}{3}\left(J^* - J(\phi_t)\right) + \frac{8\alpha_t}{3}\|\nabla J(\phi_t) - d_t\| \\
&\quad + L_J\|\phi_{t+1} - \phi_t\|^2 + \frac{\alpha_t}{3}\epsilon', \\
J^* - J(\phi_{t+1}) &\leq J^* - J(\phi_t) - \frac{\alpha_t\sqrt{\mu'}}{3}\left(J^* - J(\phi_t)\right) + \frac{8\alpha_t}{3}\|\nabla J(\phi_t) - d_t\| \\
&\quad + L_J\|\phi_{t+1} - \phi_t\|^2 + \frac{\alpha_t}{3}\epsilon', \\
\delta_{\phi_{t+1}} &\leq \left(1 - \frac{\alpha_t\sqrt{\mu'}}{3}\right)\delta_{\phi_t} + \frac{8\alpha_t}{3}\|\nabla J(\phi_t) - d_t\| \\
&\quad + L_J\|\phi_{t+1} - \phi_t\|^2 + \frac{\alpha_t}{3}\epsilon',
\end{aligned}$$

where we denote $\delta_{\phi_t} = J^* - J(\phi_t)$.

We begin by considering the update rule for δ_{ϕ_t} at timestep t:

$$\delta_{\phi_{t+1}} \leq \left(1 - \frac{\alpha_t\sqrt{\mu'}}{3}\right)\delta_{\phi_t} + \frac{8\alpha_t}{3}\|\nabla J(\phi_t) - d_t\| + L_J\|\phi_{t+1} - \phi_t\|^2 + \frac{\alpha_t}{3}\epsilon'. \tag{11}$$

Define $\beta_t = \frac{8\alpha_t}{3}\|\nabla J(\phi_t) - d_t\| + \frac{\alpha_t}{3}\epsilon' + L_J\|\phi_{t+1} - \phi_t\|^2$. Using the recursive inequality, we expand for T steps:

$$\delta_{\phi_T} \leq \delta_{\phi_{t_0}} \prod_{t=t_0}^{T-1}(1 - a\alpha_t) + \sum_{t=t_0}^{T-1}\beta_t \prod_{k=t+1}^{T-1}(1 - a\alpha_k), \tag{12}$$

where $a = \frac{\sqrt{\mu'}}{3}$. Let $\alpha_t = \frac{2}{t+\tau}$.

$$\delta_{\phi_T} \leq \frac{(t_0+\tau-1)^2}{(T+\tau-1)^2}\delta_{\phi_{t_0}} + \frac{1}{T^2}\sum_{t=t_0}^{T-1}(t+\tau)^2\beta_t. \tag{13}$$

The term β_t can be expanded and summed:

$$\frac{1}{T^2}\sum_{t=t_0}^{T-1}(t+\tau)^2\beta_t = \frac{1}{T^2}\sum_{t=t_0}^{T-1}\left[\frac{8}{3}\alpha_t\|\nabla J(\phi_t) - d_t\| + \frac{\alpha_t}{3}\epsilon' + L_J\alpha_t^2\right]. \tag{14}$$

$$\delta_{\phi_T} \leq \frac{(t_0+\tau-1)^2}{(T+\tau-1)^2}\delta_{\phi_{t_0}} + \frac{1}{T^2}\sum_{t=t_0}^{T-1}(t+\tau)^2\left[\frac{8}{3}\alpha_t\|\nabla J(\phi_t) - d_t\| + \frac{\alpha_t}{3}\epsilon' + L_J\alpha_t^2\right]. \tag{15}$$

Breaking it into individual components: - The term $\frac{8}{3}\alpha_t\|\nabla J(\phi_t) - d_t\|$ scales as $\frac{1}{T}$. - The term $\frac{\alpha_t}{3}\epsilon'$ scales as $\frac{\epsilon'}{T}$. - The term $L_J\alpha_t^2$ scales as $\frac{L_J}{T^2}$.

With the step size choice $\alpha_t = \frac{2}{a(t+\tau)}$ where $a = \frac{\sqrt{\mu'}}{3}$, $\tau = 1$, and $t_0 = 1$:

$$\delta_{\phi_T} \leq \frac{1}{T^2}\delta_{\phi_1} + \frac{1}{T^2}\sum_{t=1}^{T-1}(t+1)^2\left[\frac{8}{3}\alpha_t\|\nabla J(\phi_t) - d_t\| + \frac{\alpha_t}{3}\epsilon' + L_J\alpha_t^2\right] \tag{16}$$

Expand each component of β_t:

- **Gradient error term**:

$$\frac{8}{3}\alpha_t\|\nabla J(\phi_t) - d_t\| = \frac{16}{\sqrt{\mu'}(t+1)}\|\nabla J(\phi_t) - d_t\|$$

- **Bias term** ϵ':

$$\frac{2\epsilon'}{3\sqrt{\mu'}T^2}\sum_{t=1}^{T-1}(t+1) = \mathcal{O}\left(\frac{\epsilon'}{T}\right)$$

- **Squared step size term**:

$$\frac{36L_J}{\mu' T^2}\sum_{t=1}^{T-1}1 = \mathcal{O}\left(\frac{1}{T}\right)$$

Combine all terms and applying lemma B.4:

$$J^* - J(\phi_t) \leq \underbrace{\frac{1}{T^2}\left(J^* - J(\phi_1)\right)}_{\mathcal{O}(1/T^2)} + \frac{1}{T^2}\sum_{t=1}^{T-1}(t+1)\frac{16}{\sqrt{\mu'}}\left[\frac{8}{3}\|\nabla J(\phi_t) - d_t\|\right] + \underbrace{\mathcal{O}\left(\frac{\epsilon'}{T}\right) + \mathcal{O}\left(\frac{1}{T}\right)}_{\mathcal{O}(1/T)} \tag{17}$$

$$\leq \mathcal{O}\left(\frac{1}{T}\right) + \frac{1}{T^2}\sum_{t=1}^{T-1}(t+1)\frac{16}{\sqrt{\mu'}}\left[\|\nabla J(\phi_t) - d_t\|\right], \tag{18}$$

$$\leq \mathcal{O}\left(\frac{1}{T}\right) + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right), \tag{19}$$

□

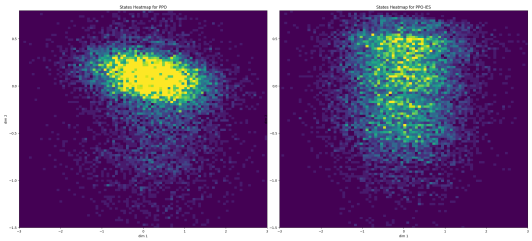


Figure 5: Heatmap of state visitation density of PPO (left) and PPO-IES (right). Yellow denotes high density and Blue denotes low density.

C EXPERIMENTS

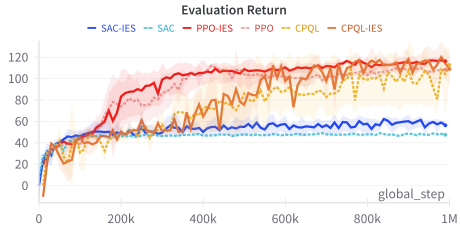
C.1 EXPLORATION VIA IES

In this section, we visualize the sampled policy trajectories under IES and standard Monte Carlo sampling. Multiple trajectories are collected from policies trained using standard PPO and those trained with IES. The trajectories are presented as state visitation heatmaps. To enable comparison, we transform the high-dimensional state space using PCA (Principal Component Analysis) and discretize the space to compute visitation counts.

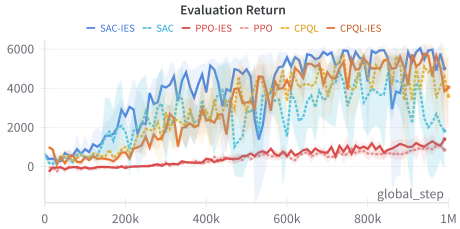
As shown in Figure 5, the state visitation patterns for PPO exhibit a concentration around a narrow region, while IES leads to more diverse and spread-out state visitation, suggesting enhanced exploration and trajectory diversity. This indicates that IES encourages the agent to explore a wider portion of the state space compared to standard Monte Carlo sampling.

C.2 ADDITIONAL EXPERIMENTS

Additionally, we provide the training curve for `Mujoco` v4 environments and the evaluation return table for `IssacGym` environments. It is shown that IES performs better than other exploration strategy in many environments.



(a) Swimmer-v4



(b) Ant-v4

Figure 6: Performance on MuJoCo.

Table 3: Performance comparison of different methods across `IsaacGym` environments. A-Hand denotes AllegroHand, S-Hand denotes ShadowHand, B-Balance denotes BallBalance, F-C denotes FrankaCabinet, Franka-C-T denotes FrankaCabinetTimeout.

Method	A-Hand	S-Hand	Ball-B	Franka-C	Humanoid	Franka-C-T	Anymal
PPO	200	15	**345**	1799	2641	824	28
PPO-RND	201	**35**	317	1644	824	972	31
PPO-RLE	192	18	306	1334	886	1370	37
PPO-IES (Ours)	**204**	25	328	**2084**	**2897**	**2081**	**39**