UNCERTAIN REWARDS BRING MORE POSSIBILITIES: BIDIRECTIONAL REWARD NOISE FOR EXPLORATION IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep reinforcement learning (DRL) relies on reward-driven policy updates, yet sufficient exploration remains a critical challenge for discovering high-quality policies. Due to the tendency to pursue high rewards, agents are prone to becoming trapped around rewarding states encountered early in training, hindering exploration of potentially more valuable regions and leading to premature convergence. Existing exploration methods commonly introduce additional stochasticity or intrinsic incentives to encourage broader exploration. However, stochasticity alone lacks explicit directional control, while intrinsic rewards often require additional modeling components. In this paper, we propose Bidirectional Reward Noise (BiRN), a lightweight exploration mechanism that directly perturbs training rewards. By injecting bidirectional noise, BiRN achieves two complementary effects: negative noise mitigates excessive attraction toward dominant rewarding transitions, while positive noise strengthens incentives for transitions with positive learning potential. Extensive experiments show that BiRN consistently improves final performance across diverse RL benchmarks with minimal additional overhead, with particularly strong gains in challenging sparse-reward settings.

1 INTRODUCTION

Deep Reinforcement Learning (DRL) optimizes policies through continuous interaction with the environment, which has demonstrated strong decision-making capabilities in various applications such as robotic control, games, and autonomous driving (Kiran et al., 2022). To acquire effective strategies, DRL agents need to continuously explore the environment and learn from trial and error to accumulate experience. Therefore, sufficient exploration is crucial for discovering high-quality policies. However, when rewarding states are encountered early in the learning process, their repeated reinforcement may cause the agent to concentrate prematurely on a limited region of the state space, preventing it from further exploring potentially more valuable states. Consequently, the repeated reinforcement of early rewards can drive premature exploitation of suboptimal regions, posing a key challenge to effective exploration.

To improve exploration, researchers have developed various mechanisms that introduce additional stochasticity or exploratory incentives into the learning process. Two representative directions are randomized exploration and intrinsic reward-based exploration. Randomization-based exploration methods introduce perturbations into the learning process, for instance through actions, policy parameters, or value estimates (Fortunato et al., 2018; Plappert et al., 2018; Kveton et al., 2019; Eberhard et al., 2023; Mahankali et al., 2024). Such stochasticity can effectively broaden exploration, but randomness alone does not provide explicit directional control over reward-driven learning. Another widely studied direction is intrinsic reward-based exploration (Ramesh et al., 2022; Kim et al., 2023), which augments environmental rewards with additional signals to encourage exploratory behaviors. Many intrinsic reward methods rely on additional novelty estimators, representation models, or auxiliary objectives, which introduce extra modeling and computational requirements. This motivates a simple question: *Can exploration be regulated directly through temporary perturbations of the reward signal?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Motivated by this perspective, we investigate reward perturbation as a lightweight exploration mechanism that directly modifies the training reward to regulate reward-driven updates. To examine whether reward perturbation itself can facilitate exploration, we first study a simple selective random reward perturbation scheme, referred to as Selective Random Reward Noise (SrRN). Although this simple random perturbation can broaden exploration, our analysis and empirical observations reveal a key limitation: the perturbation is directionally blind. In particular, positive perturbations may further amplify already dominant rewarding transitions, reinforcing the agent's tendency to remain concentrated around early reward peaks. Inspired by this phenomenon, we propose **Bidirectional Reward Noise (BiRN)**, a directionally controlled reward perturbation strategy. BiRN leverages noise in two directions: negative noise tempers currently dominant rewarding transitions, while positive noise strengthens incentives toward alternative transitions. Through this bidirectional adjustment, BiRN mitigates premature concentration around early reward peaks and facilitates continued exploration of alternative regions.

We evaluate BiRN across diverse single-agent benchmarks, including discrete-control Atari games, continuous-control IsaacGym tasks, and sparse-reward MiniGrid tasks. BiRN can be readily integrated into existing DRL algorithms and consistently improves final performance across the evaluated tasks. The gains are particularly pronounced in challenging sparse-reward environments where early rewarding regions can induce premature exploitation. We further evaluate BiRN in several cooperative multi-agent environments, providing preliminary evidence that the same reward perturbation principle can extend beyond single-agent settings. Our main contributions are as follows.

- We formally analyze random reward perturbation as an exploration strategy and reveal its directional blindness, whereby random perturbations may inadvertently reinforce early reward attractors.
- We propose Bidirectional Reward Noise (BiRN), a directionally controlled reward perturbation mechanism that tempers currently dominant rewarding transitions while strengthening incentives for potentially promising states.
- Extensive experiments across diverse benchmarks and reinforcement learning algorithms demonstrate that BiRN consistently improves final performance with minimal additional computational overhead.

2 PRELIMINARY

Reinforcement learning (RL) has emerged as a powerful paradigm for solving sequential decision-making problems in complex environments. The standard RL framework formalizes the interaction between an agent and its environment as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space and $\mathcal{R}(s_t, a_t) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. At each timestep t, the agent perceives the environment state $s_t \in \mathcal{S}$ and executes an action $a_t \in \mathcal{A}$. The transition probability function $\mathcal{P}(s_{t+1}|s_t, a_t)$ governs the progression from the current state to the next state s_{t+1} upon the agent's action. The agent receives an immediate reward $r_t = \mathcal{R}(s_t, a_t)$. The agent's objective is to learn a policy π that maximizes the expected cumulative reward $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$, where $\gamma \in [0, 1)$ is a discount factor. In practice, various RL algorithms learn a state-action value function $Q(s_t, a_t) = \mathbb{E}_\pi[G_t|s_t, a_t]$ or state value function $V(s_t) = \mathbb{E}_\pi[G_t|s_t]$ to approximate the expected cumulative discounted reward $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$.

Exploration in reinforcement learning Exploration in reinforcement learning refers to an agent's behavior for gathering information about insufficiently understood regions of the state-action space (Hao et al., 2024). Since the optimal policy is unknown during training, an agent must balance exploring uncertain alternatives with exploiting its current knowledge. Insufficient exploration often leads to premature convergence to suboptimal policies, especially when early updates repeatedly concentrate the agent around a limited set of rewarding states.

3 MOTIVATION: INVESTIGATING RANDOM REWARD PERTURBATION

In the early stages of training, an agent often quickly focuses on the first few high-reward transitions it encounters, limiting exploration of the broader state space. A natural idea is to perturb rewards to

introduce stochasticity into the learning targets and thereby reduce overly aggressive early commitment. To explore this idea, we begin with a simple random reward perturbation scheme to examine whether reward noise can facilitate exploration.

3.1 Selective Random Reward Noise

To encourage exploration, we inject stochasticity into the rewards of selected transitions during training, which we call *Selective Random Reward Noise* (SrRN). Given a training batch $\mathcal{B} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^{|\mathcal{B}|}$, we compute the one-step bootstrap target $y_i = r_i + \gamma V(s_i')$ and the TD error $\delta_i = y_i - V(s_i) = r_i + \gamma V(s_i') - V(s_i)$. A larger absolute TD error $|\delta_i|$ typically indicates that the corresponding state is poorly understood by the current value function, suggesting it may benefit from additional exploration. Thus, we treat $|\delta_i|$ as an indicator of potential underexploration and prioritize transitions with large $|\delta_i|$ for stochastic perturbation. Conceptually, one could define a set of candidate transitions using a threshold x as $U = \{s_i | |\delta_i| \geq x\}$. In practice, however, a fixed threshold x would require environment-wise tuning. Therefore, we select the top-K fraction of transitions with the largest $|\delta_i|$ values in the batch, where $K \in (0, 1]$ is a selection ratio.

For each selected transition $i \in \mathcal{I}_K(\mathcal{B})$, we inject Gaussian reward noise:

$$\tilde{r}_i = r_i + \lambda \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, 1), \tag{1}$$

where $\lambda \in (0, 1]$ is the noise scale. The resulting bootstrap target is $\tilde{y}_i = \tilde{r}_i + \gamma V(s_i') = y_i + \lambda \epsilon_i$.

This perturbation is straightforward to analyze. Since the injected noise has zero mean, the perturbed target is unbiased with respect to the noise: $\mathbb{E}_{\epsilon_i}[\tilde{y}_i] = y_i$. At the same time, the perturbation increases the variance of the target by $\mathrm{Var}_{\epsilon_i}(\tilde{y}_i) = \lambda^2$. This added stochasticity directly encourages exploration around the selected underexplored states.

3.2 Limitation of Random Perturbation

Despite its simplicity, the random perturbation scheme described above has a limitation: it provides no directional control. For a selected transition, the perturbed TD error is $\tilde{\delta}_i = \tilde{y}_i - V(s_i) = \delta_i + \lambda \epsilon_i$. Depending on the sign of ϵ_i, the perturbation may either amplify or attenuate the original TD error. In particular, when the perturbation has the opposite sign to δ_i, it can weaken the corrective effect of the update and even reverse its direction. For example, if $\delta_i < 0$, then $\Pr(\tilde{\delta}_i > 0) = 1 - \Phi\left(\frac{|\delta_i|}{\lambda}\right)$, where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function. Therefore, although SrRN introduces useful stochasticity, it provides no control over whether the perturbation strengthens or weakens the current learning update. This limitation suggests that, to guide exploration more effectively, reward perturbation should move beyond purely random noise and incorporate explicit directional control.

4 Bidirectional Reward Noise Injection

To this end, we propose Bidirectional Reward Noise (BiRN) to introduce explicit directional control over reward perturbation. Unlike purely random noise, BiRN determines the perturbation direction based on the relative target values within a batch. Such directional control is particularly valuable in sparse-reward environments, where the empirical reward landscape often exhibits sharp local peaks. Random positive perturbations can make early high-reward regions even more attractive, increasing the risk of premature exploitation. BiRN aims to counteract this tendency by tempering the attraction of currently favored regions while sustaining incentives to explore promising alternative regions, reducing the risk of premature convergence to suboptimal policies.

4.1 Bidirectional Perturbation Rule

BiRN perturbs rewards only for transitions with positive TD errors, i.e., $\delta_i > 0$, which indicate that the current value estimate underpredicts the bootstrap target and is therefore subject to an upward update. Let $\bar{y}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} y_j$ denote the batch mean of the original targets, and $\mathcal{B}^+ = \{i \in \mathcal{B} : \delta_i > 0\}$ denote the set of positively reinforced transitions. For each transition, BiRN perturbs the

reward as $\tilde{r}_i = r_i + \lambda\eta_i$, where

$$\eta_i = -\mathbf{1}\{\delta_i > 0\}\,\mathrm{sgn}(y_i - \bar{y}_{\mathcal{B}})\,\epsilon_i^+, \quad \epsilon_i^+ \sim \mathcal{N}^+(0,1). \tag{2}$$

$\mathcal{N}^+(0,1)$ denotes the half-normal distribution and $\mathrm{sgn}(\cdot)$ denotes the sign function with the tie-breaking convention $\mathrm{sgn}(0) = -1$. The perturbed target is $\tilde{y}_i = \tilde{r}_i + \gamma V(s_i') = y_i + \lambda\eta_i$. The perturbation rule produces two complementary directional effects, as illustrated in Figure 1.

Negative noise for escaping local optima For transitions with $\delta_i > 0$ and $y_i > \bar{y}_{\mathcal{B}}$, the sign term $\mathrm{sgn}(y_i - \bar{y}_{\mathcal{B}})$ is positive, yielding a negative perturbation: $\tilde{r}_i = r_i - \lambda\epsilon_i^+$. These transitions have relatively high targets that continue to drive upward value updates. In sparse-reward environments, such reinforcement can concentrate the policy around early reward peaks before potentially more valuable regions are explored. As conceptually illustrated in Figure 1, negative noise lowers the effective height of these peaks in the training reward landscape, tempering their attraction to reduce the risk of premature convergence to suboptimal policies.

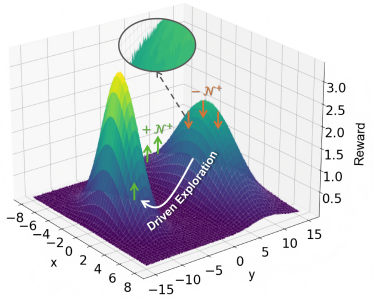


Figure 1: Conceptual illustration of BiRN's directional reward perturbation.

Positive noise for encouraging exploration Conversely, for transitions that exhibit upward learning potential $\delta_i > 0$ but remain below the current batch average $y_i \le \bar{y}_{\mathcal{B}}$, BiRN injects positive perturbation: $\tilde{r}_i^+ = r_i + \lambda \cdot \epsilon_i^+$. BiRN treats these transitions as promising alternatives to currently favored regions. By optimistically boosting the training rewards, BiRN amplifies upward value updates and encourages exploration beyond early reward peaks.

4.2 Directional Effect Analysis

To characterize the directional effect, we first consider perturbations with the batch and current value estimates held fixed. We analyze the signed deviation of the perturbed target from the batch mean, i.e., $(\tilde{y}_i - \bar{y}_{\mathcal{B}})\,\mathrm{sgn}(y_i - \bar{y}_{\mathcal{B}})$.

Proposition 1. *For any $i \in \mathcal{B}^+$ with $y_i \neq \bar{y}_{\mathcal{B}}$, the following hold conditioned on $\mathcal{B}$:* $\mathbb{E}[(\tilde{y}_i - \bar{y}_{\mathcal{B}})\,\mathrm{sgn}(y_i - \bar{y}_{\mathcal{B}}) \mid \mathcal{B}] = |y_i - \bar{y}_{\mathcal{B}}| - \lambda\sqrt{\frac{2}{\pi}}$, *and* $\mathrm{Var}_{\eta_i}(\tilde{y}_i \mid \mathcal{B}) = \lambda^2\left(1 - \frac{2}{\pi}\right)$.

The proof is deferred to Appendix C. Proposition 1 shows that the bidirectional perturbation reduces the expected signed outward deviation of each perturbed target from the batch-average target by $\lambda\sqrt{2/\pi}$. This directional adjustment lowers above-average targets and raises below-average targets among transitions with positive TD errors. Compared with the zero-mean perturbations in SrRN, BiRN introduces a directional shift in the learning targets. At the same noise scale λ, BiRN also has a lower conditional perturbation variance than SrRN on perturbed transitions.

Perturbation control In implementation, we additionally clip the sampled noise to $[-1, 1]$ before scaling and exponentially decay λ for both SrRN and BiRN. If the original rewards are bounded, the perturbed rewards remain bounded up to a controlled additive term. Moreover, for a fixed batch and value estimates, the induced target perturbation is bounded in magnitude by λ and vanishes as $\lambda \to 0$. The perturbations modify the training targets, while the original environmental return remains the task objective and evaluation criterion. Appendix D provides the derivation.

The above discussion is presented in terms of state-value TD errors for clarity. The same principle can be directly extended to action-value estimation by using the TD error of $Q(s_i, a_i)$ and the corresponding bootstrap target to determine reward perturbations. In this way, BiRN can also guide exploration through state–action value updates. More generally, both SrRN and BiRN can be readily integrated into DRL algorithms that rely on Bellman-style updates, reusing their existing learning signals without introducing additional trainable components or modifying the policy and value network architectures.

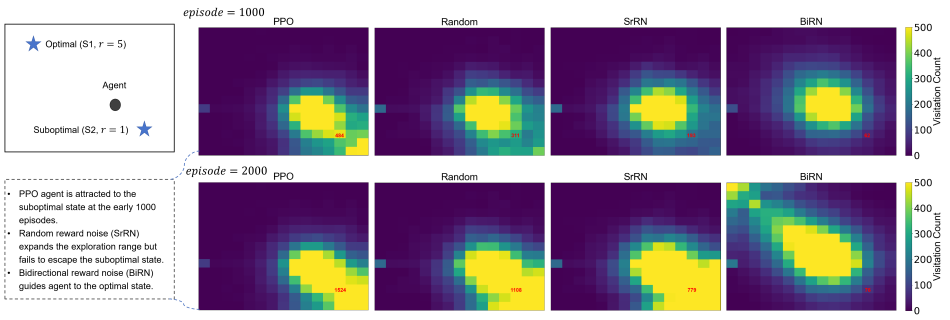
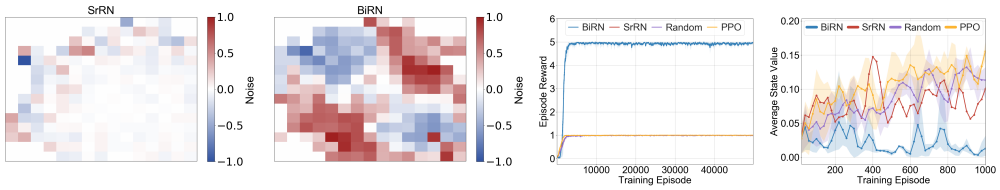


Figure 2: The sparse-reward 15×15 grid exploration environment and state visitation comparison.



(a) (b) (c) (d)

Figure 3: Noise analysis in the grid exploration environment. (a)-(b): Average noise applied to each state during the training of 1000 episodes. (c) The reward curves of different algorithms. (d) The average evaluated state value of the suboptimal states during the early 1000 episodes.

5 Experiments

In this section, we first illustrate the impact of our method on exploration in a grid environment, then demonstrate its effectiveness across both single-agent and multi-agent scenarios. Our experiments cover a variety of environments, including discrete-control Atari games, continuous-control Isaac-Gym environments, and three multi-agent tasks. We integrate our method with PPO for single-agent tasks, MADDPG (Lowe et al., 2017) and QMIX(Rashid et al., 2018) for multi-agent tasks. We run experiments with 5 random seeds and report the average results with 95% confidence intervals. Pseudocodes and implementation code are provided in Appendix A. For training hyperparameters, $K = 50\%$ is fixed for SrRN, and detailed settings for λ are reported in Appendix Table 7.

5.1 Illustration example: How does Reward Noise work?

We examine how reward noise affects exploration in the sparse-reward 15×15 grid environment shown in Figure 2. The environment contains an optimal state S1 with reward 5 and a suboptimal state S2 with reward 1, while all other states yield zero reward. Each episode starts from a fixed position closer to S2 and lasts for at most 25 steps. All variants use the same PPO backbone. For all perturbation variants, the initial noise scale is $\lambda = 1$ and decays throughout training; SrRN uses $K = 50\%$. The Random baseline removes TD-based selection and directional control while retaining the same clipping bound and scale schedule. We record the cumulative reward noise applied to each state and the visitation frequency throughout training, and visualize the averaged results over multiple runs using heatmaps.

State visitation Figure 2 compares state visitation during early training. During the first 1k episodes, PPO, Random, SrRN, and BiRN visit S2 484, 311, 193, and 62 times, respectively. Random and SrRN explore more broadly than PPO but remain centered around S2 after 2k episodes, showing that undirected noise does not sufficiently weaken the attraction of an early reward region. BiRN instead reduces S2 revisits and shifts exploration toward S1, tempering premature reinforcement while sustaining the search for higher reward. We further evaluate this early-stage exploration behavior against entropy-tuned PPO and PPO-Count in Appendix E.1.

Impact of noise on state exploration Figure 3(a)–(b) shows the cumulative reward noise applied to each state during the first 1k training episodes. For SrRN, the cumulative noise is similar and

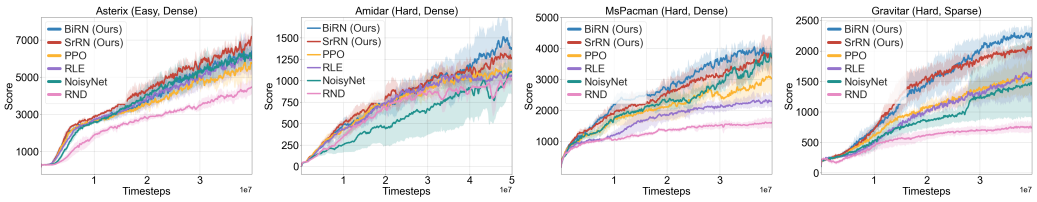


Figure 4: Learning curves for the selected four Atari games with different settings of easy/hard exploration and dense/sparse rewards.

close to zero across most states, indicating that its random perturbations are broadly distributed over potentially under-explored states. However, the similar cumulative effect across states provides little directional guidance for exploration. In contrast, BiRN applies negative noise around the two high-reward states, reducing their continued attraction, while applying more positive noise to distant zero-reward regions. These results show that BiRN makes reward-noise exploration more directional, helping the agent move beyond currently favored regions and continue searching for higher reward.

Estimated state values We perform 10 evaluation runs every 1000 training episodes, recording both the episode reward and the estimated state value of the suboptimal state S2 during training. Figure 3(c) shows that BiRN rapidly escapes the suboptimal policy and reaches the optimal return. As shown in Figure 3(d), we visualize the average estimated value of the suboptimal state during the first 1000 episodes. BiRN suppresses the value estimate of S2 early in training, preventing premature convergence to inferior policies. In contrast, the other methods keep relatively high estimates for S2, making it harder to escape local optima. SrRN further exhibits large fluctuations, with several peaks even higher than PPO, which may occasionally reinforce suboptimal behaviors.

5.2 Effectiveness in Single-Agent Environments

To evaluate the performance under different RL environments, we implement the proposed reward noise-driven methods on PPO and A2C and consider RND (Burda et al., 2019), NoisyNet (Fortunato et al., 2018), VCSE (Kim et al., 2023), RE3 (Seo et al., 2021) and RLE (Mahankali et al., 2024) as baselines. We perform empirical evaluations on discrete control Atari 2600 games, continuous control IsaacGym and sparse-reward MiniGrid benchmarks to investigate the impact of reward noise.

5.2.1 Results on Atari

The Atari games (Bellemare et al., 2013) encompass a wide variety of environments. Following common practices in Atari, the agent perceives a stack of the most recent four 84×84 grayscale frames as inputs and takes discrete actions available on the Atari gamepad (Mnih et al., 2015). We utilize the standard PPO hyperparameters in (Burda et al., 2019), with the learning rate uniformly set to 3e-4, consistent with the setting in (Mahankali et al., 2024). We select a variety of Atari games based on the categorization in (Oh et al., 2018), including one easy-exploration task and three hard-exploration tasks featuring dense or sparse rewards. Figure 4 shows the game score of different algorithms. Overall, the results on four Atari games demonstrate that the two proposed reward noise injection mechanisms effectively enhance the performance of the baseline PPO. Compared to other baselines, they also exhibit superior performance. SrRN and BiRN perform comparably to other algorithms in environments with easy exploration but demonstrate advantages in the three hard exploration tasks, especially in the sparse-reward Gravitar task. This indicates that the two proposed reward noise exploration strategies effectively promote exploration efficiency in various discrete control tasks. Moreover, as detailed in Table 4 of Appendix E.2, SrRN and BiRN introduce an average additional runtime of only around 1.05% compared to PPO in Atari environments.

5.2.2 Results on IsaacGym

To demonstrate that SrRN and BiRN can improve upon PPO in continuous control tasks, we conduct experiments in IsaacGym benchmark suite (Makoviychuk et al., 2021), which contains numerous continuous control tasks. We train our algorithms with standard PPO hyperparameters in IsaacGym provided in CleanRL (Huang et al., 2022). We follow the parameter settings used in RLE for all the

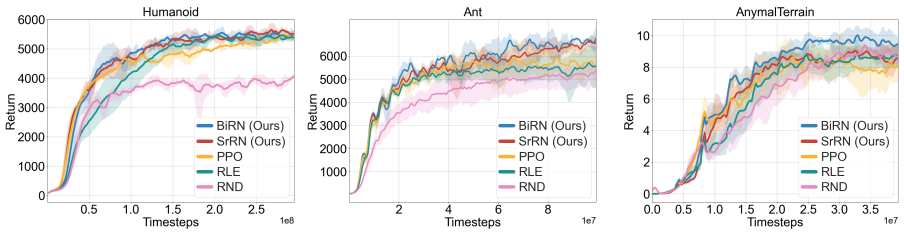


Figure 5: Results on IsaacGym environments. Our methods outperform the other baselines.

Table 1: Average test return and runtime per $1e4$ training steps in MiniGrid. ↑ denotes performance improvement over its baselines.

Method	**Empty-25x25**		**MultiRoom-N2S4**		**Fetch-8x8-N3**	
	Return	*Time(s)*	*Return*	*Time(s)*	*Return*	*Time(s)*
A2C	0.12 ± 0.01	3.10	0.67 ± 0.07	5.37	0.34 ± 0.03	4.62
A2C+BiRN	0.74 ± 0.42 ↑	3.25	0.74 ± 0.01 ↑	5.58	0.38 ± 0.03 ↑	4.65
RE3	0.14 ± 0.02	7.43	0.70 ± 0.02	8.43	0.34 ± 0.02	7.03
RE3+BiRN	0.76 ± 0.40 ↑	7.96	0.76 ± 0.03 ↑	8.53	0.37 ± 0.02 ↑	7.49
VCSE	0.54 ± 0.51	6.67	0.80 ± 0.01	8.38	0.36 ± 0.02	7.52
VCSE+BiRN	0.98 ± 0.01 ↑	7.50	0.81 ± 0.01 ↑	8.43	0.38 ± 0.02 ↑	7.63

algorithms, where we use reward normalization as in RLE. Figure 5 demonstrates that SrRN and BiRN achieve higher average returns than all baselines, which indicates their effectiveness in environments with high-dimensional continuous state and action spaces. Together, the positive results in the Atari and IsaacGym benchmarks highlight the applicability and effectiveness of SrRN and BiRN in enhancing exploration for both discrete and continuous control tasks.

5.2.3 Results on Sparse-Reward MiniGrid

To further assess BiRN in challenging sparse-reward settings, we evaluate it on the MiniGrid benchmark (Chevalier-Boisvert et al., 2018). We compare it against RE3 and VCSE, two recent A2C-based methods that augment exploration with intrinsic state-entropy rewards. For a fair comparison, we inject reward noise into the raw reward signals of A2C, RE3, and VCSE during training. Our implementation is built on the official VCSE codebase. As shown in Table 1, BiRN consistently improves final performance across all baselines, suggesting that bidirectional reward perturbation is complementary to intrinsic reward design rather than tied to a specific backbone. This gain is achieved without noticeable computational overhead, indicating that BiRN is a lightweight exploration mechanism. Additional results are provided in Appendix E.7.

5.3 Can BiRN Generalize to Multi-Agent Settings?

To assess whether BiRN can extend beyond single-agent settings, we conduct evaluation in three multi-agent tasks. Exploration in multi-agent environments is inherently more complex, since each agent must adapt not only to the environment but also to the evolving behaviors of other agents. Although BiRN does not explicitly model inter-agent coordination, its perturbation of rewards can still affect policy updates and thereby influence coordination dynamics.

5.3.1 M-step matrix game

We first consider an M-step matrix game (Mahajan et al., 2019) with $M = 12$, where the optimal strategy requires two agents to coordinate over all M steps and yields the maximum payoff of 15. Any single misstep results in a suboptimal payoff, creating a narrow exploration bottleneck. We integrate BiRN into the value targets of QMIX and compare it with vanilla QMIX, MAVEN (Mahajan et al., 2019), and ICES (Li et al., 2024). As shown in Figure 6, QMIX gets stuck at the suboptimal payoff of 12, whereas QMIX-BiRN reaches the maximum payoff of 15. This suggests that the directional perturbations can help prevent early overcommitment to rewarded yet suboptimal trajectories. While MAVEN and ICES rely on specialized architectural designs for exploration,

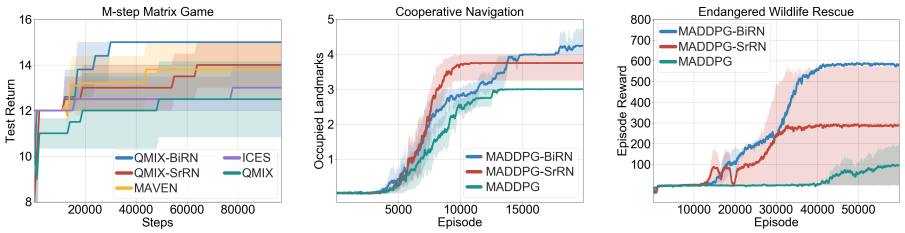


Figure 6: Results on multi-agent environments.

neither reaches the optimal payoff within the training budget. By contrast, BiRN attains the optimal strategy as a generic add-on without requiring multi-agent-specific architectural modifications.

5.3.2 Multi-agent particle environments

We further evaluate BiRN on two sparse-reward cooperative multi-agent tasks derived from the multi-agent particle environment (MPE) (Lowe et al., 2017): cooperative navigation and endangered wildlife rescue. In cooperative navigation, five agents must simultaneously cover five landmarks without collisions within limited interaction steps. In the more challenging endangered wildlife rescue task, which follows the multi-stage setting in (Chen et al., 2022), two rescue agents need to save all wounded animals according to injury severity. Figure 6 shows that BiRN achieves better final performance than MADDPG, suggesting that it can facilitate the discovery of underexplored targets. Although many existing MARL exploration methods are designed to induce diverse collaborative behaviors, these results indicate that BiRN can serve as a lightweight complementary mechanism for encouraging exploration in cooperative multi-agent tasks.

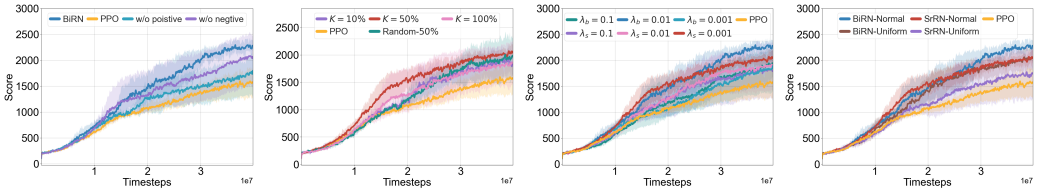
5.4 Discussions

Ablation on BiRN To investigate the role of positive and negative noise in BiRN, we conduct ablative experiments by removing positive noise (w/o positive) and negative noise (w/o negative) respectively. Figure 7(a) shows that both ablations degrade performance, indicating that both directionally injected positive and negative noise in BiRN are critical for performance improvement. Positive noise encourages exploration of promising underexplored regions, while negative noise helps avoid local suboptimality and further optimizes the final performance. Although negative noise is effective for escaping local optima, the agent still struggles to discover better policies without positive guidance for underexplored states.

The impact of different top-K in SrRN We validate the impact of adding noise to different proportions of samples in the Gravitar, where the previously reported results are under the setting of $K = 50\%$. We evaluate K at $\{10\%, 50\%, 100\%\}$ respectively, where $K = 100\%$ indicating that noise is added to all samples. We additionally implement *Random-50%*, which randomly selects 50% of the samples to assess the effectiveness of focusing on underexplored states in SrRN. Figure 7(b) shows that adding noise to specific proportions of samples can enhance exploration performance. SrRN with $K = 50\%$ outperforms both *Random-50%* and $K = 100\%$ (which is equivalent to *Random-100%*), suggesting that selectively adding noise to a portion of underexplored states is more effective for promoting agent exploration than random or universal perturbation.

The impact of initial λ Figure 7(c) shows the performance of BiRN and SrRN on Gravitar with initial noise scales $\lambda \in \{0.001, 0.01, 0.1\}$, denoted by λ_b and λ_s, respectively. Both methods achieve higher final mean scores than PPO across this two-order-of-magnitude range, although the magnitude of improvement varies across settings. The initial scales used in our experiments are selected through a coarse search, with the selected values reported in Table 7.

Different reward noise distribution To investigate the influence of using Gaussian noise, we sample noise from uniform distribution $\mathcal{U}(-1, 1)$. Figure 7(d) show that uniform reward noise also outperforms PPO, which highlights the potential of our approaches to improve exploration.



(a) (b) (c) (d)

Figure 7: Results in the sparse-reward Gravitar environment: (a) Ablation on BiRN; (b) and (c) are analysis results of hyperparameter sensitivity for K and initial λ. (d) Learning curves of BiRN and SrRN under normal/uniform noise.

6 Related Work

Randomized exploration Randomized perturbation is a straightforward strategy for encouraging exploration in RL. Many RL algorithms employ simple heuristic exploration strategies such as adding noise to actions or randomly sampling actions (e.g., ϵ-greedy). Related approaches inject noise into the parameter space (Rückstieß et al., 2010; Fortunato et al., 2018) or encourage stochastic policies through entropy maximization (Haarnoja et al., 2018). These approaches primarily promote exploration by increasing behavioral stochasticity, without explicitly controlling how reward-driven learning signals are strengthened or attenuated. Another line of approach is randomized value function methods (Kveton et al., 2019; Ishfaq et al., 2021; Li et al., 2022; Ishfaq et al., 2024) inspired by Thompson Sampling (THOMPSON, 1933; Russo et al., 2018), which introduce randomness into value estimation to induce temporally extended exploration. The recent RLE (Mahankali et al., 2024) introduces feature-related randomized reward functions in PPO to promote trajectory diversity. RLE uses structured reward randomization, but does not explicitly control perturbation direction based on current learning signals. In contrast, BiRN uses current learning signals to determine the perturbation direction, thereby reducing the risk that reward randomization further reinforces already dominant reward regions. Randomized value-function methods have also been extended to parallel MDP settings (Hsu et al., 2024), while their role in partially observable cooperative multi-agent exploration remains less studied.

Intrinsic reward-based exploration Intrinsic reward-based exploration methods have been extensively studied to provide additional feedback for encouraging exploration. Existing approaches construct intrinsic rewards from signals such as novelty, prediction error, or state visitation (Bellemare et al., 2016; Sekar et al., 2020; Ramesh et al., 2022), and have demonstrated strong performance in sparse-reward tasks. Constructing these signals often involves additional estimators or learned representations. BiRN also guides exploration through training rewards, but constructs perturbations from existing TD errors and relative target values. Intrinsic rewards also play an important role in multi-agent reinforcement learning, where they encourage exploration and cooperative behaviors (Du et al., 2019; Liu et al., 2021; 2023; Li et al., 2024). Coordination-oriented methods use intrinsic incentives to shape interactions among agents, whereas BiRN modulates existing reward signals to encourage continued exploration. Combining reward perturbations with explicit incentives for diverse cooperative behaviors is a promising direction for future work.

7 Conclusion

In this paper, we propose BiRN, a directionally controlled reward perturbation strategy for exploration in DRL. Our investigation of selective random reward noise motivates controlling how perturbations strengthen or attenuate learning signals. BiRN uses current TD information and relative target values to temper the attraction of currently favored regions while encouraging continued exploration. Experiments across single-agent benchmarks and cooperative multi-agent tasks demonstrate performance improvements, with clearer advantages over SrRN in challenging sparse-reward settings. By reusing existing learning signals without introducing additional trainable components or changing network architectures, BiRN offers a lightweight implementation that can be integrated into existing DRL algorithms and combined with intrinsic-reward methods.

AI USE STATEMENT

Generative AI tools were used solely to assist with language editing and writing refinement. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed experimental settings, implementation details, hyperparameters, and reward-noise configurations in Appendix G. The reinforcement learning environments and baseline implementations used in this work are publicly available. Our source code, configurations, and experiment scripts are available in an anonymous repository[1].

ETHICS STATEMENT

We affirm that all authors have read and adhere to the ICLR Code of Ethics. Our work is based on standard reinforcement learning benchmarks and simulated environments and does not involve human or animal subjects, sensitive personal data, or identifiable privacy risks.

REFERENCES

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.*, 47:253–279, 2013. doi: 10.1613/JAIR.3912. URL https://doi.org/10.1613/jair.3912.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1471–1479, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations*. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Xinning Chen, Xuan Liu, Shigeng Zhang, Bo Ding, and Kenli Li. Goal consistency: An effective multi-agent cooperative method for multistage tasks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 172–178. ijcai.org, 2022. doi: 10.24963/IJCAI.2022/25. URL https://doi.org/10.24963/ijcai.2022/25.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. 2018.

Yali Du, Lei Han, Meng Fang, Ji Liu, Tianhong Dai, and Dacheng Tao. LIIR: learning individual intrinsic reward in multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4405–4416, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/07a9d3fed4c5ea6b17e80258dee231fa-Abstract.html.

Onno Eberhard, Jakob J. Hollenstein, Cristina Pinneri, and Georg Martius. Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=hQ9V5QN27eS.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *6th International Conference*

[1] https://anonymous.4open.science/r/Reward-Noise-driven-Exploration

on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=rywHCPkAW.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

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*, 35(7):8762–8782, 2024. doi: 10.1109/TNNLS.2023.3236361. URL https://doi.org/10.1109/TNNLS.2023.3236361.

Hao-Lun Hsu, Weixin Wang, Miroslav Pajic, and Pan Xu. Randomized exploration in cooperative multi-agent reinforcement learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/8839556afa4efe2a7dc5aae5e0a22fd4-Abstract-Conference.html.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G. M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *J. Mach. Learn. Res.*, 23:274:1–274:18, 2022. URL https://jmlr.org/papers/v23/21-1342.html.

Haque Ishfaq, Qiwen Cui, Viet Nguyen, Alex Ayoub, Zhuoran Yang, Zhaoran Wang, Doina Precup, and Lin Yang. Randomized exploration in reinforcement learning with general value function approximation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4607–4616. PMLR, 2021. URL http://proceedings.mlr.press/v139/ishfaq21a.html.

Haque Ishfaq, Qingfeng Lan, Pan Xu, A. Rupam Mahmood, Doina Precup, Anima Anandkumar, and Kamyar Azizzadenesheli. Provable and practical: Efficient exploration in reinforcement learning via langevin monte carlo. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=nfIAEJFiBZ.

Dongyoung Kim, Jinwoo Shin, Pieter Abbeel, and Younggyo Seo. Accelerating reinforcement learning with value-conditional state entropy exploration. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6530db249c161fe9254db2667453952c-Abstract-Conference.html.

B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Kumar Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.*, 23(6):4909–4926, 2022. doi: 10.1109/TITS.2021.3054625. URL https://doi.org/10.1109/TITS.2021.3054625.

Branislav Kveton, Csaba Szepesvári, Mohammad Ghavamzadeh, and Craig Boutilier. Perturbed-history exploration in stochastic multi-armed bandits. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 2786–2793. ijcai.org, 2019. doi: 10.24963/IJCAI.2019/386. URL https://doi.org/10.24963/ijcai.2019/386.

Xinran Li, Zifan Liu, Shibo Chen, and Jun Zhang. Individual contributions as intrinsic exploration scaffolds for multi-agent reinforcement learning. In *Forty-first International Conference on Machine Learning*. OpenReview.net, 2024. URL https://openreview.net/forum?id=zCmMkWK4Ly.

Ziniu Li, Yingru Li, Yushun Zhang, Tong Zhang, and Zhi-Quan Luo. Hyperdqn: A randomized exploration method for deep reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=X0nrKAXu7g-.

Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and Du Zhang. Lazy agents: A new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21937–21950. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23ac.html.

Iou-Jen Liu, Unnat Jain, Raymond A. Yeh, and Alexander G. Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *ICML*, volume 139, pp. 6826–6836, 2021. URL http://proceedings.mlr.press/v139/liu21j.html.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017. URL http://papers.nips.cc/paper/7217-multi-agent-actor-critic-for-mixed-cooperative-competitive-environments.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. MAVEN: multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, pp. 7611–7622, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/f816dc0acface7498e10496222e9db10-Abstract.html.

Srinath Mahankali, Zhang-Wei Hong, Ayush Sekhari, Alexander Rakhlin, and Pulkit Agrawal. Random latent exploration for deep reinforcement learning. In *Forty-first International Conference on Machine Learning*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Y9qzwNlKVU.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/28dd2c7955ce926456240b2ff0100bde-Abstract-round2.html.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *ICML*, pp. 3875–3884, 2018. URL http://proceedings.mlr.press/v80/oh18b.html.

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. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=ByBAl2eAZ.

Aditya A. Ramesh, Louis Kirsch, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Exploring through random curiosity with general value functions. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural*

Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/76e57c3c6b3e06f332a4832ddd6a9a12-Abstract-Conference.html.

Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML*, pp. 4292–4301, 2018.

Thomas Rückstieß, Frank Sehnke, Tom Schaul, Daan Wierstra, Yi Sun, and Jürgen Schmidhuber. Exploring parameter space in reinforcement learning. *Paladyn J. Behav. Robotics*, 1 (1):14–24, 2010. doi: 10.2478/S13230-010-0002-4. URL https://doi.org/10.2478/s13230-010-0002-4.

Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Found. Trends Mach. Learn.*, 11(1):1–96, 2018. doi: 10.1561/2200000070. URL https://doi.org/10.1561/2200000070.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8583–8592. PMLR, 2020. URL http://proceedings.mlr.press/v119/sekar20a.html.

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9443–9454. PMLR, 2021. URL http://proceedings.mlr.press/v139/seo21a.html.

WILLIAM R THOMPSON. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 12 1933. ISSN 0006-3444. doi: 10.1093/biomet/25.3-4.285. URL https://doi.org/10.1093/biomet/25.3-4.285.

A ALGORITHM

This section presents the pseudocode for the two reward perturbation schemes discussed in the main paper: Selective Random Reward Noise (SrRN) and Bidirectional Reward Noise (BiRN), shown in Algorithm 1 and Algorithm 2, respectively. The code is available in a anonymous Github [2].

Algorithm 1 Selective Random Reward Noise (SrRN)

Require: Initial policy parameters θ, standard normal distribution $\mathcal{N}(0,1)$, noise scale λ, selection ratio $K \in (0,1]$
Ensure: Trained policy parameters θ
repeat
  Initialize environment state
  for $t = 0, \ldots, T-1$ **do**
    Agent interacts with environment to collect experience (s_t, a_t, r_t, s_{t+1})
  end for
  Compute TD errors: $\delta_t = r_t + \gamma V_\theta(s_{t+1}) - V_\theta(s_t)$
  Sample noise $\epsilon_t \sim \mathcal{N}(0,1)$ for all t
  Select the top-K transitions with largest $|\delta_t|$
  for each selected transition j **do**
    Inject random noise: $\tilde{r}_j \leftarrow r_j + \lambda \cdot \epsilon_j$
  end for
  for each unselected transition j **do**
    $\tilde{r}_j \leftarrow r_j$ #No perturbation
  end for
  Update policy parameters θ using the underlying RL algorithm with perturbed rewards $\{\tilde{r}_j\}$
until Convergence

Algorithm 2 Bidirectional Reward Noise (BiRN)

Require: Initial policy parameters θ, half-normal distribution $\mathcal{N}^+(0,1)$, noise scale λ
Ensure: Trained policy parameters θ
repeat
  Initialize environment state
  for $t = 0, \ldots, T$ **do**
    Agent interacts with environment to collect experience (s_t, a_t, r_t, s_{t+1})
  end for
  Sample a mini-batch of transitions $\{(s_j, a_j, r_j, s'_j)\}_{j=1}^{N_T}$ and compute TD errors $\{\delta_j\}_{j=1}^{N_T}$
  Compute target values $y_j = r_j + \gamma V_\theta(s'_j)$ and their average $\bar{V}_{\text{target}} = \frac{1}{N_T}\sum_j y_j$
  Sample positive noise $\epsilon_j \sim \mathcal{N}^+(0,1)$ for each j
  for each transition j **do**
    if $\delta_j > 0$ and $y_j > \bar{V}_{\text{target}}$ **then**
      Inject negative noise: $\tilde{r}_j \leftarrow r_j - \lambda \cdot \epsilon_j$
    else if $\delta_j > 0$ and $y_j \leq \bar{V}_{\text{target}}$ **then**
      Inject positive noise: $\tilde{r}_j \leftarrow r_j + \lambda \cdot \epsilon_j$
    else
      $\tilde{r}_j \leftarrow r_j$ #No perturbation
    end if
  end for
  Update policy parameters θ using the underlying RL algorithm with perturbed rewards $\{\tilde{r}_j\}$
until Convergence

[2] https://anonymous.4open.science/r/Reward-Noise-driven-Exploration

B Effect of Gaussian reward perturbation in SrRN

For each selected transition $i \in \mathcal{I}_K(\mathcal{B})$, SrRN perturbs the reward as $\tilde{r}_i = r_i + \lambda\epsilon_i, \epsilon_i \sim \mathcal{N}(0,1)$, which yields the perturbed bootstrap target $\tilde{y}_i = y_i + \lambda\epsilon_i, \epsilon_i \sim \mathcal{N}(0,1)$. Conditioned on the current batch $\mathcal{B}$ and value function V, since $\mathbb{E}[\epsilon_i] = 0$, we have $\mathbb{E}_{\epsilon_i}[\tilde{y}_i] = \mathbb{E}_{\epsilon_i}[y_i + \lambda\epsilon_i] = y_i + \lambda\mathbb{E}[\epsilon_i] = y_i$, and $\mathrm{Var}_{\epsilon_i}(\tilde{y}_i) = \mathrm{Var}_{\epsilon_i}(y_i + \lambda\epsilon_i) = \lambda^2\mathrm{Var}(\epsilon_i) = \lambda^2$. Therefore, the Gaussian reward perturbation used in SrRN preserves the conditional mean of the bootstrap target while increasing its conditional variance by λ^2.

C Proof of Proposition 1

For any $i \in \mathcal{B}^+$, the noise term is $\eta_i = -\operatorname{sgn}(y_i - \bar{y}_\mathcal{B})\,\epsilon_i^+$, where $\epsilon_i^+ \sim \mathcal{N}^+(0,1)$, and thus the value target becomes $\tilde{y}_i = y_i + \lambda\eta_i$. We have

$$\begin{aligned}(\tilde{y}_i - \bar{y}_\mathcal{B})\operatorname{sgn}(y_i - \bar{y}_\mathcal{B}) &= \big((y_i - \bar{y}_\mathcal{B}) + \lambda\eta_i\big)\operatorname{sgn}(y_i - \bar{y}_\mathcal{B}) \\ &= |y_i - \bar{y}_\mathcal{B}| - \lambda\epsilon_i^+.\end{aligned}$$

Taking conditional expectation given $\mathcal{B}$ and using $\mathbb{E}[\epsilon_i^+] = \sqrt{\frac{2}{\pi}}$ for $\epsilon_i^+ \sim \mathcal{N}^+(0,1)$ yields

$$\mathbb{E}[(\tilde{y}_i - \bar{y}_\mathcal{B})\operatorname{sgn}(y_i - \bar{y}_\mathcal{B}) \mid \mathcal{B}] = |y_i - \bar{y}_\mathcal{B}| - \lambda\sqrt{\frac{2}{\pi}}. \tag{3}$$

For the variance, Since y_i and $\operatorname{sgn}(y_i - \bar{y}_\mathcal{B})$ are deterministic given $\mathcal{B}$, we have

$$\mathrm{Var}(\tilde{y}_i \mid \mathcal{B}) = \mathrm{Var}(y_i + \lambda\eta_i \mid \mathcal{B}) = \lambda^2\,\mathrm{Var}(\eta_i) = \lambda^2\,\mathrm{Var}(\epsilon_i^+). \tag{4}$$

The variance of the standard half-normal distribution is $1 - 2/\pi$. Hence, we obtain

$$\mathrm{Var}(\tilde{y}_i \mid \mathcal{B}) = \lambda^2\left(1 - \frac{2}{\pi}\right). \tag{5}$$

D Bounded and Vanishing Reward Perturbation

This section establishes the boundedness and vanishing properties of the reward perturbations introduced by BiRN and their effects on bootstrap targets. At each training iteration, the original and perturbed targets are compared using the same batch and value estimates. At training iteration n, recall that BiRN perturbs the reward r_i for a transition $i \in \mathcal{B}_n$ as:

$$\tilde{r}_i^{(n)} = r_i + \lambda_n \cdot \mathrm{clip}(\eta_i^{(n)}, -1, 1), \tag{6}$$

where $\eta_i^{(n)} = -\mathbf{1}\{\delta_i^{(n)} > 0\}\operatorname{sgn}(y_i^{(n)} - \bar{y}_{\mathcal{B}_n})\epsilon_i^+$ with $\epsilon_i^+ \sim \mathcal{N}^+(0,1)$. Here, $\delta_i^{(n)}$ denotes the TD error, and $\lambda_n \in [0,1]$ is the decaying noise scale.

For any iteration n and transition $i \in \mathcal{B}_n$, the reward perturbation and the resulting target perturbation $\Delta y_i^{(n)} = \tilde{y}_i^{(n)} - y_i^{(n)}$ satisfy:

$$|\tilde{r}_i^{(n)} - r_i| \le \lambda_n \quad \text{and} \quad |\tilde{y}_i^{(n)} - y_i^{(n)}| \le \lambda_n. \tag{7}$$

Furthermore, the conditional mean target shift vanishes as $n \to \infty$:

$$\left|\mathbb{E}_{\epsilon^+}\left[\tilde{y}_i^{(n)} - y_i^{(n)} \mid \mathcal{B}_n\right]\right| \to 0 \quad \text{as} \quad \lambda_n \to 0. \tag{8}$$

Proof. By the definition of the clipped perturbation, we have:

$$|\tilde{r}_i^{(n)} - r_i| = \lambda_n \cdot \left|\mathrm{clip}(\eta_i^{(n)}, -1, 1)\right|. \tag{9}$$

Since $|\mathrm{clip}(\eta_i^{(n)}, -1, 1)| \le 1$, it follows that

$$|\tilde{r}_i^{(n)} - r_i| = \lambda_n \left|\mathrm{clip}(\eta_i^{(n)}, -1, 1)\right| \le \lambda_n. \tag{10}$$

For the bootstrap target $y_i^{(n)} = r_i + \gamma V(s_i')$, since the perturbation only affects the immediate reward term, the target difference is:

$$\tilde{y}_i^{(n)} - y_i^{(n)} = (\tilde{r}_i^{(n)} + \gamma V(s_i')) - (r_i + \gamma V(s_i')) = \tilde{r}_i^{(n)} - r_i, \tag{11}$$

which implies $|\tilde{y}_i^{(n)} - y_i^{(n)}| \leq \lambda_n$. Finally, by Jensen's inequality, the magnitude of the conditional expectation is bounded by the expectation of the magnitude:

$$\left| \mathbb{E}_{\epsilon^+} \left[\tilde{y}_i^{(n)} - y_i^{(n)} \mid \mathcal{B}_n \right] \right| \leq \mathbb{E}_{\epsilon^+} \left[|\tilde{y}_i^{(n)} - y_i^{(n)}| \mid \mathcal{B}_n \right] \leq \lambda_n. \tag{12}$$

As $\lambda_n \to 0$ by the decay schedule, the conditional mean target shift vanishes asymptotically. □

Remark 1 (Preservation of reward boundedness)**.** *If the original rewards satisfy $|r_i| \leq R_{\max}$, the perturbed rewards satisfies $|\tilde{r}_i^{(n)}| \leq R_{\max} + \lambda_n \leq R_{\max} + 1$. Thus, clipping maintains boundedness up to a controlled additive term.*

E Addition Results

E.1 Additional exploration baselines in the grid task

We further compare BiRN with count-based exploration and different entropy regularization strengths in the same grid task. PPO-Count adds an exploration bonus $r^+(s) = \alpha/\sqrt{N(s)}$ to the environmental reward during training, where $N(s)$ denotes the state visit count. We search over $\alpha \in \{0.1, 0.5, 1.0\}$ and report the best-performing setting, $\alpha = 0.5$, while keeping the PPO architecture and underlying hyperparameters unchanged. We also evaluate PPO with entropy coefficients $\beta \in \{0.001, 0.01, 0.1\}$, where $\beta = 0.01$ is the default setting. The Random baseline uses the same noise scale, decay schedule, and clipping bound as BiRN.

Table 2: Returns in the grid exploration task at different training checkpoints.

Method	1k	2k	3k	50k
PPO	0.53 ± 0.01	0.96 ± 0.00	0.99 ± 0.00	1.00 ± 0.00
PPO-Count	0.20 ± 0.07	2.36 ± 0.75	4.49 ± 0.22	4.99 ± 0.03
BiRN	0.18 ± 0.11	3.84 ± 0.35	4.86 ± 0.11	4.99 ± 0.02

Comparison with count-based exploration Table 2 reports returns at four training checkpoints over five random seeds. PPO initially obtains a higher mean return by exploiting the nearby suboptimal state S2, but its return remains near 1 throughout subsequent training. Both PPO-Count and BiRN achieve near-optimal returns by 50k episodes. BiRN obtains higher mean returns at 2k and 3k episodes, showing faster progress toward the optimal reward during this period.

Table 3: Exploration statistics during the first 1k training episodes in the grid task. S2 is the nearby suboptimal reward state.

Method	State coverage (%)	Visits to S2
PPO ($\beta = 0.001$)	75.56	648
PPO ($\beta = 0.01$)	75.78	484
PPO ($\beta = 0.1$)	76.44	706
Random	76.44	311
SrRN	81.78	193
PPO-Count	97.07	142
BiRN	94.67	62

State coverage and visits to the suboptimal state Table 3 compares state coverage and visits to S2 during the first 1k training episodes. Across the tested entropy coefficients, PPO exhibits similar state coverage and ultimately remains at the suboptimal return of 1. Both PPO-Count and BiRN substantially expand state coverage relative to PPO. Compared with PPO-Count, BiRN has slightly

lower coverage (94.67% versus 97.07%) but approximately 56% fewer visits to S2 (62 versus 142). Together with the returns in Table 2, these results show that BiRN's faster early learning is accompanied by fewer visits to the suboptimal reward state while maintaining broad state coverage.

E.2 Computation efficiency

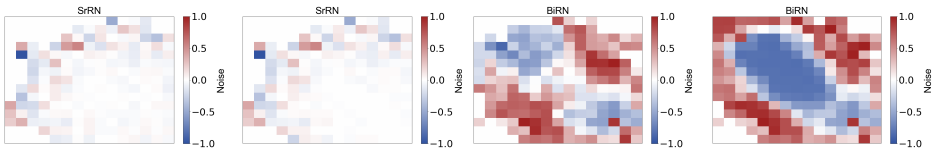
As shown in Table 4, we provide the average processing time required for running $1e7$ steps across different Atari games, along with the percentage increase in runtime for SrRN and BiRN relative to PPO. SrRN and BiRN introduce an average additional runtime of only 0.975% and 1.13% compared to PPO, respectively. This indicates that SrRN and BiRN can effectively enhance exploration while introducing only a very small additional computational cost.

Table 4: Average Runtime (minutes: seconds) per 1e7 Steps in the Atari games and the percentage increase in runtime of SrRN and BiRN relative to PPO.

	SrRN	BiRN	PPO
Amidar	22:56 (+0.3%)	22:58 (+0.5%)	22:51
Gravitar	26:23 (+0.5%)	26:19 (+0.3%)	26:15
MrsPacman	25:05 (+1.5%)	25:22(+2.6%)	24:43
Asterix	26:30 (+1.6%)	26:22(+1.1%)	26:05

E.3 Noise applied to states in grid exploration task

In Figure 8, we provide the average noise added by SrRN and BiRN at each state in the initial 1000 episodes, as well as throughout the entire training process (totaling 50k episodes). It can be observed that the trend in the final stage of training is similar to the early stages. Although SrRN can guide the agent to explore states further away from the suboptimal by adding noise, the random noise fails to help the agent escape the suboptimal. In contrast, BiRN effectively avoids getting trapped in suboptimal states by applying bidirectional noise, promoting more thorough exploration by the agent.



(a) 1k episodes (b) 50k episodes (c) 1k episodes (d) 50k episodes

Figure 8: The total sum of noise applied to each state dring the initial 1k episodes and the final 50k episodes.

E.4 State visitations in the grid exploration task

Figure 9 provides detailed state visitation maps at different stages of training. By 2000 episodes, BiRN's visitation extends to the optimal state, while PPO and SrRN remain concentrated around the suboptimal region. At 50000 episodes, BiRN shows substantial visitation along the route to the optimal state, whereas PPO and SrRN continue to concentrate on the suboptimal region.

E.5 Additional ablation results on K

We investigate the impact of parameter K on performance in a multi-agent cooperative navigation environment. As shown in Figure 10(a), injecting reward noise over different proportions of selected transitions consistently improves the performance of MADDPG, although the magnitude of the gain

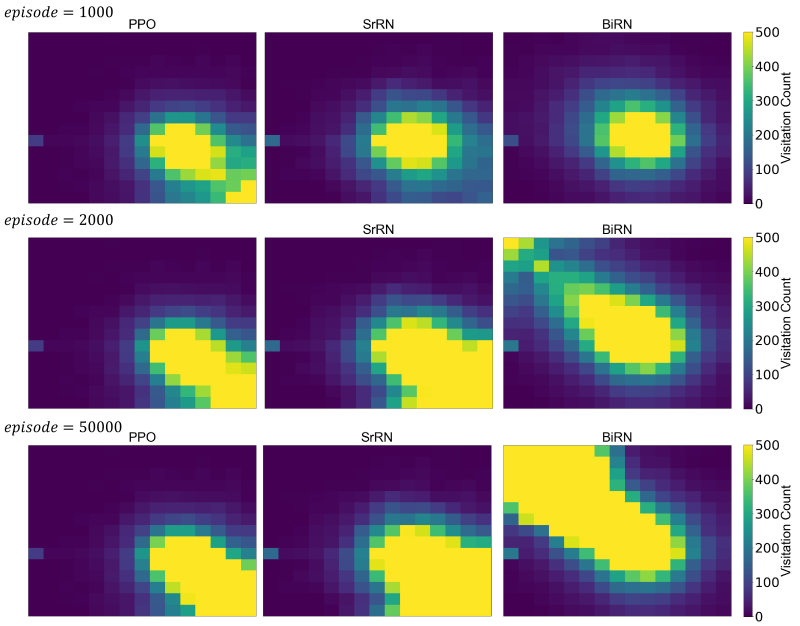
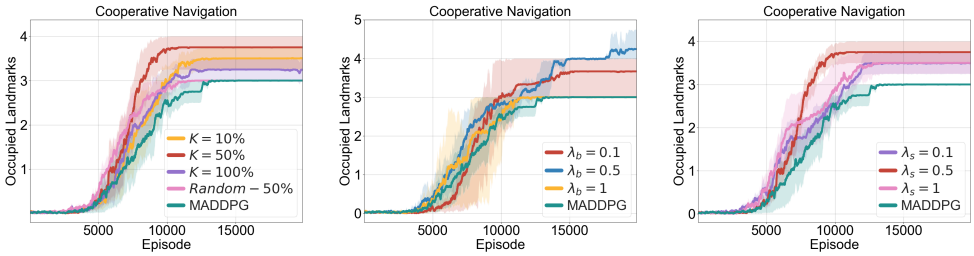


Figure 9: The state visitations of different algorithms in the grid exploration environment.



(a) Different K (b) Different λ_b (c) Different λ_s

Figure 10: Analysis on the parameter K, λ_b for BiRN and λ_s for SrRN.

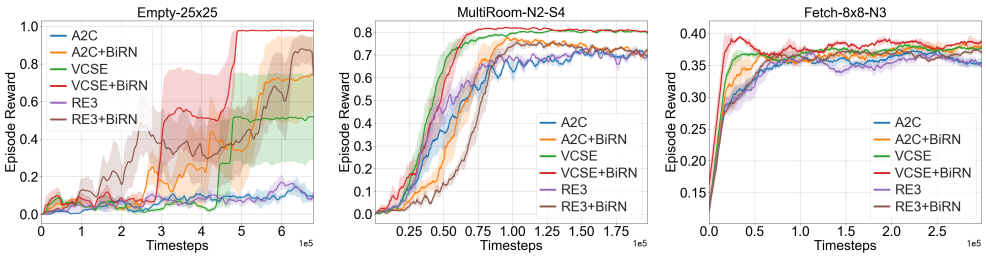
varies across choices of K. These results further support the role of reward perturbation as an effective exploration mechanism in multi-agent tasks, while indicating that the method is robust to the choice of K.

E.6 Additional ablation results on initial λ

We further examine the effect of the initial noise scale λ in a multi-agent cooperative navigation environment. For both BiRN and SrRN, we evaluate $\lambda \in \{0.1, 0.5, 1\}$. As shown in Figure 10(b)-(c), different noise scales lead to different performance gains, while all tested settings outperform MADDPG. This result suggests that the proposed reward perturbation remains effective within the tested range of noise magnitudes. Although the preferred value of λ can vary with the reward scale and environment, a coarse search is sufficient to obtain consistent improvements in this setting.

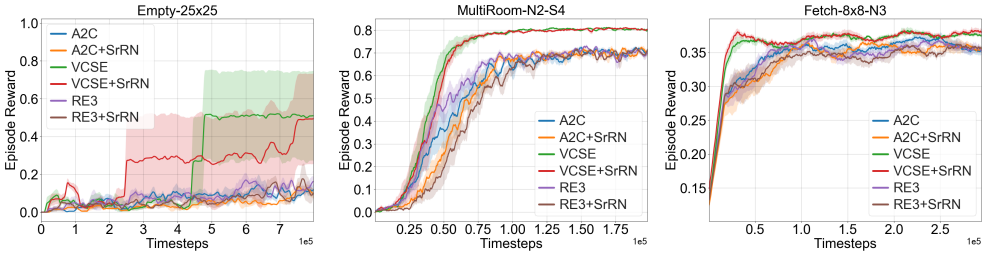
E.7 Results in MiniGrid

We evaluate our method on three navigation tasks from MiniGrid benchmark:Empty-25x25, MultiRoom-N2S4 and Fetch-8x8. We use the original tasks without any modification for our experiments. For comparison, we specifically consider A2C, RE3 and VCSE as baselines. RE3 trains A2C agent with state entropy exploration where the intrinsic reward is obtained using the representations from a random encoder. VCSE maximizes the value-conditional state entropy and considers



(a) (b) (c)

Figure 11: Reward curves of algorithms that combined BiRN in MiniGrid.



(a) (b) (c)

Figure 12: Reward curves of algorithms that combined SrRN in MiniGrid.

the visited states with similar value estimates for computing the intrinsic bonus. To show that our methods can be applied to existing methods to enhace their exploration efficiency, we apply SrRN and BiRN to A2C, RE3 and VCSE, respectively. Figure 11 presents the episode rewards of algorithms integrated with BiRN, which consistently improves both the baselines' learning efficiency and final episode rewards. In contrast, Figure 12 demonstrates that SrRN fails to boost exploration efficiency in these sparse-reward environments.

Early exploration behavior We further examine exploration behavior in MiniGrid Empty-25x25. Table 5 reports state coverage and visit frequency during the first 10k environment steps, together with test success rates. Adding BiRN to A2C and RE3 increases mean state coverage and lowers mean visit frequency, alongside higher mean test success rates. For VCSE, mean coverage and test success change little, while mean visit frequency increases. These results provide additional behavioral evidence for BiRN's exploration effect, with clearer improvements observed for A2C and RE3 in this environment.

F Applicability and Limitations

Exploration regimes BiRN is designed for settings where early rewarding experiences can cause premature concentration before alternative paths are sufficiently explored. Its benefits are therefore expected to be more pronounced when this tendency limits exploration. In our experiments, BiRN shows clearer advantages over SrRN in hard-exploration and sparse-reward tasks, while the two methods perform comparably on Asterix and the IsaacGym tasks. When exploration is already sufficient or early reward attractors are weak, directional perturbations may offer limited additional benefit.

Reward-scale calibration The strength of BiRN's intervention depends on the initial noise scale relative to the processed training rewards. A scale that is too small may have little exploration effect, whereas an excessively large scale may interfere with useful learning signals and hurt performance. Consequently, a single numerical setting need not transfer across tasks with different reward scales. This calibration concerns one scalar coefficient; the perturbation mechanism itself reuses existing TD errors and bootstrap targets without requiring auxiliary estimators or changes to the policy and

value networks. BiRN thus retains lightweight implementation across algorithms, while requiring task-dependent selection of the perturbation strength.

Table 5: Exploration statistics and test success rates in MiniGrid Empty-25x25. Coverage and visit frequency are measured during the first 10k environment steps.

Method	Coverage@10k	Visit frequency@10k	Test success rate
A2C	0.35 ± 0.05	47.73 ± 7.56	0.23 ± 0.03
A2C + BiRN	0.43 ± 0.05	37.84 ± 4.33	0.61 ± 0.45
RE3	0.35 ± 0.03	46.60 ± 4.80	0.31 ± 0.07
RE3 + BiRN	0.56 ± 0.17	31.05 ± 9.76	0.63 ± 0.43
VCSE	0.46 ± 0.09	35.92 ± 5.86	0.76 ± 0.49
VCSE + BiRN	0.47 ± 0.25	43.45 ± 23.78	0.77 ± 0.46

G Detailed Experimental Settings

All experiments are performed on NVIDIA GeForce RTX A6000 graphics card. To counteract the randomness from a statistical perspective, we conduct multiple trials using different seeds. We provide detailed parameter settings in Table 8 for the grid exploration task, Table 9 for Atari games, Table 10 for IsaacGym environments. And we provide the hyperparameters of SrRN and BiRN in different environments in Table 7. For multi-agent cooperative navigation environment, we follow the settings of original MADDPG. For M-step matrix game, we follow the settings in MAVEN.

Reward preprocessing and noise configuration Reward perturbations are applied after the reward preprocessing used by the underlying baseline. In IsaacGym, all compared algorithms use reward normalization, and noise is added to the normalized rewards. We clip the sampled noise to $[-1, 1]$ before multiplying it by λ, so λ bounds the absolute reward perturbation in the units of the processed training rewards. We choose the initial scale relative to these rewards through a coarse search over $\{0.001, 0.01, 0.1, 0.5, 1\}$ and exponentially decay it throughout training. Table 7 reports the selected initial scales.

M-step matrix game The initial state exhibits non-monotonic properties, with zero rewards triggering termination; distinguishing states are located at the terminal ends, and there exist $M - 2$ intermediate states. Figure 13 shows the m-step matrix game with $M = 12$, the optimal strategy involves selecting the top-left joint action initially and the bottom-right action finally, yielding an optimal total payoff of $M + 3$. Additionally, the initial state's nonmonotonicity provides inertia against switching the policy to the other direction. AsMincreases, discovering the optimal policy grows progressively more challenging.

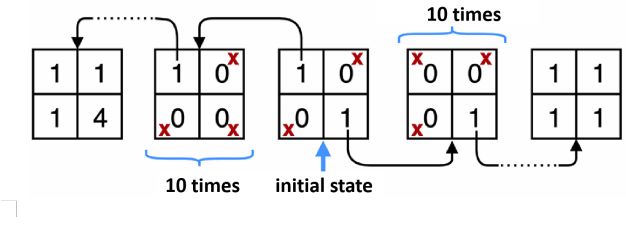


Figure 13: M-step matrix game for $M = 12$.

Cooperative navigation We adopt a sparse-reward version of multi-agent cooperative navigation environment with continuous state and action spaces (Lowe et al., 2017). This task requires five agents to simultaneously cover five landmarks without collisions under limited interaction steps. Each agent has only local observations while sharing a global reward. Each agent gets a positive reward c^2, where c denotes the number of occupied landmarks, and a negative reward -0.1 for each collision.

Endangered wildlife rescue We consider a sparse-reward cooperative task derived from the multi-agent particle environment (MPE) (Lowe et al., 2017), following the multi-stage setting in (Chen et al., 2022). In this task, two rescue agents cooperate to save wounded animals with different injury severities. Each target requires both agents to arrive and rescue it simultaneously. The task is challenging because the agents must not only coordinate their movements but also prioritize targets according to injury severity. The payoff matrix of agent i is defined in Table 6. At each step, if both agents capture and save the same target a or b simultaneously, they both receive a reward of 100 or 10, respectively. Otherwise, if agent i captures target j while its partner captures target k, agent i receives $r(j,k)$. If agent i holds target j but its partner does not arrive within 3 steps, both agents receive a penalty of $r(j,3)$. All agents observe the relative positions of the other agents and targets.

Table 6: The payoff matrix of the rescue agent i at each step.

		Partner		
		catch a	catch b	on the road
Agent i	catch a	100	1	-10
	catch b	-10	10	-5
	on the road	-10	-5	0

H BROADER IMPACTS

Although randomized exploration techniques have been extensively investigated, the study of reward noise-driven exploration remains relatively under-explored. This paper proposes two novel reward noise-driven methods and shows their ability to enhance reinforcement learning performance across a variety of tasks. This work holds promise for extending into more intricate reinforcement learning domains and stimulating further research interest in the field of reward noise.

Table 7: Hyperparameters for SrRN and BiRN in different evaluated environments.

	λ_b	λ_s
Asterix	0.01	0.01
Amidar	0.001	0.001
MsPacman	0.1	0.1
Gravitar	0.01	0.001
Humanoid	0.001	0.01
Ant	0.001	0.01
Anymalterrain	0.01	0.1
MiniGrid	0.5	0.5
Cooperative Navigation	0.5	0.5
M-matrix Game	0.5	0.5
Grid Exploration	1	1

Table 8: Hyperparameters and network architectures for grid exploration experiments.

Parameter	Value
Total Episode	50,000
Optimizer	RMSprop
Learning Rate	0.00004
Discount Rate	0.99
Generalized Advantage Estimation	0.95
Clipping Coefficient	0.2
Entropy Loss Weight	0.01
Discount Rate	0.99
Value Loss Weight	0.5
Policy Network Architecture	MLP (64,64,5)
Value Network Architectures	MLP (64,64,1)
Network Activation	ReLU

Table 9: Hyperparameters and network architectures for Atari experiments.

Parameter	Value
PPO	
Optimizer	Adam
Learning Rate	0.0003
Adam Epsilon	0.00001
Parallel Workers	128
Steps per Batch	128
Discount Rate	0.99
Generalized Advantage Estimation	0.95
Minibatches per Epoch	4
Epochs per Training Step	4
Clipping Coefficient	0.1
Entropy Loss Weight	0.01
Discount Rate	0.99
Value Loss Weight	0.5
Gradient Norm Bound	0.5
Use Advantage Normalization	True
Use Clipped Value Loss	True
Policy Network Architecture	CNN + MLP (256,448,448,18)
Value Network Architectures	CNN + MLP (256,448,448,1)
Network Activation	ReLU
NoisyNet	
Initial σ	0.017
Entropy Loss Weight	0
RND	
Intrinsic Reward Coefficient	1.0
Extrinsic Reward Coefficient	2.0
Drop Probability	0.25
Discount Rate	0.999
Entropy Loss Weight	0.001
Intrinsic Discount Rate	0.99
Predictor Network Architecture	CNN + MLP (512,512,512)
Target Network Architecture	CNN + MLP (512)
Network Activation	LeakyReLU
RLE	
Intrinsic Reward Coefficient	0.01
Latent Vector Dimension	8
Latent Vector Resample Frequency	1280
Feature Network Update Rate τ	0.005
Feature Network Architecture	CNN MLP (256,448,8)

Table 10: Hyperparameters and network architectures for Isaacgym experiments.

Parameter	Value
PPO	
Optimizer	Adam
Learning Rate	0.0026
Adam Epsilon	0.000001
Steps per Batch	16
Discount Rate	0.99
Generalized Advantage Estimation	0.95
Minibatches per Epoch	2
Epochs per Training Step	4
Clipping Coefficient	0.2
Entropy Loss Weight	0.0
Discount Rate	0.99
Value Loss Weight	2.0
Gradient Norm Bound	1.0
Use Advantage Normalization	True
Use Clipped Value Loss	False
Policy Network Architecture	MLP (256,256,256)
Value Network Architecture	MLP (256,256,256,1)
Network Activation	Tanh
Reward Scale	1.0
RLE	
Intrinsic Value Loss Weight	0.5
Intrinsic Reward Coefficient	0.01
Latent Vector Dimension	32
Latent Vector Resample Frequency	16
Learning Rate	0.0001
Feature Network Update Rate τ	0.005
Policy Network Architecture	MLP (256,256,256)
Value Network Architecture	MLP (512,512,256,1)
Feature Network Architecture	MLP (512,512,256)