Rethinking Exploration in Reinforcement Learning with Effective Metric-Based Exploration Bonus

Yiming Wang[1], **Kaiyan Zhao**[1,2], **Furui Liu**[3], **Leong Hou U**[1*]
[1]State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao SAR, China
[2]School of Computer Science, Wuhan University, Wuhan, China
[3]Zhejiang Lab, Hangzhou, China
wang.yiming@connect.um.edu.mo, zhao.kaiyan@whu.edu.cn
liufurui@zhejianglab.com, ryanlhu@um.edu.mo

Abstract

Enhancing exploration in reinforcement learning (RL) through the incorporation of intrinsic rewards, specifically by leveraging *state discrepancy* measures within various metric spaces as exploration bonuses, has emerged as a prevalent strategy to encourage agents to visit novel states. The critical factor lies in how to quantify the difference between adjacent states as *novelty* for promoting effective exploration. Nonetheless, existing methods that evaluate state discrepancy in the latent space under L_1 or L_2 norm often depend on count-based episodic terms as scaling factors for exploration bonuses, significantly limiting their scalability. Additionally, methods that utilize the bisimulation metric for evaluating state discrepancies face a theory-practice gap due to improper approximations in metric learning, particularly struggling with *hard exploration* tasks. To overcome these challenges, we introduce the **E**ffective **M**etric-based **E**xploration-bonus (EME). EME critically examines and addresses the inherent limitations and approximation inaccuracies of current metric-based state discrepancy methods for exploration, proposing a robust metric for state discrepancy evaluation backed by comprehensive theoretical analysis. Furthermore, we propose the diversity-enhanced scaling factor integrated into the exploration bonus to be dynamically adjusted by the variance of prediction from an ensemble of reward models, thereby enhancing exploration effectiveness in particularly challenging scenarios. Extensive experiments are conducted on hard exploration tasks within Atari games, Minigrid, Robosuite, and Habitat, which illustrate our method's scalability to various scenarios. The project website can be found at https://sites.google.com/view/effective-metric-exploration.

1 Introduction

Reinforcement learning (RL) has made significant strides, yielding breakthroughs across various domains, including video gaming [43], autonomous driving [65], and robotic control [1, 37]. However, for many real-world tasks, defining a dense reward function is non-trivial, yet a sparse reward function based on success or failure is directly available, which makes learning effective policies difficult, as they demand efficient exploration of the state space, highlighting exploration in sparse-reward environments as a core challenge in RL [59].

Methods that quantify the state discrepancy between adjacent steps using specific measures within different metric spaces have shown remarkable success in tasks characterized by sparse rewards. These methods, by leveraging a measure of state discrepancy as an exploration bonus, facilitate agents

[*] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

in discovering novel states. For instance, RIDE [51] employs the difference between two consecutive state embeddings, measured under L_2 norm, as an exploration bonus. Similarly, NovelD [69] introduces a bonus based on the state discrepancy, as represented by the RND [11] bonus, but under the L_1 norm. However, the effectiveness of these L_p norms-based methods is contingent on the scaling factor expressed as the episodic count term $N_{ep}(s)$, which is the number of times state s has been visited during the current episode. The episodic term becomes ineffective when each state is unique and cannot be counted, posting a significant limitation when it comes to more complex, dynamic, and noisy environments [32]. LIBERTY [63] aims to surmount this hurdle by evaluating state discrepancy through the bisimulation metric [25], which links state differences to value differences within the bisimulation metric space, thus incentivizing exploration of novel states with greater value differences and substantially enhancing learning efficiency. Nevertheless, due to the high computational cost or infeasibility of accurately computing bisimulation metrics, the approximation, and relaxations over the metric [13, 68, 63] have been used to optimize efficiency. Our analysis reveals that approximation gap in LIBERTY might break the theoretical guarantees of the bisimulation metric, potentially undermining exploration performance. Furthermore, the efficacy of bonuses based on state discrepancies declines in *hard exploration* tasks or scenarios where state differences are minimal, such as in the "Noisy-TV" [11] problem and vision-based real indoor environments [60], posing a significant constraint on the scalability of these methods.

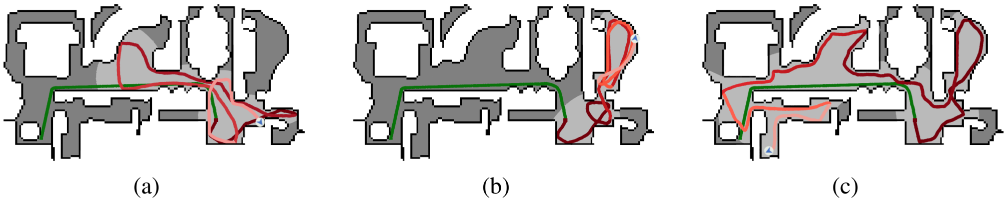


Figure 1: Trajectories of policies trained with different exploration algorithms in the real-life indoor environment. (a) episodic count-based method under L_p norm (b) bisimulation metric-based method (c) our method.

To address this aforementioned limitations, we introduce the Effective Metric-based Exploration-bonus (EME) for exploration. Our method features a more resilient metric for evaluating state differences, simultaneously eliminating approximation gap with a theoretical guarantee. EME significantly boosts learning efficiency by forging a closer connection to the value differences between states, while maintaining fidelity to its theoretical framework. Furthermore, we have developed a diversity-enhanced scaling factor to augment exploration efficacy in *hard exploration* tasks, where state differences are notably subtle. Our scaling factor dynamically adjusts the exploration bonus based on the variance of predictions from an ensemble of reward models, which is higher when agents encounter novel state space, thereby encouraging more effective exploration. As illustrated in Figure 1, we present the trajectories of policies trained using count-based state discrepancy bonus, bisimulation-based exploration bonus, and our method within a real-life indoor environment [15] (Full results in Figure 10 of Appendix C.3). Our method successfully explores a significantly larger portion of the space compared to the others, demonstrating the superior effectiveness of EME.

The main contributions of this paper are as follows. Firstly, we conduct a comprehensive analysis of the limitations inherent in current metric-based state discrepancy methods for exploration. Based on the analysis, we introduce an effective metric for evaluating the behavioral similarity between states supported by theoretical assurances. Secondly, we propose a diversity-enhanced scaling factor for exploration bonuses based on the variance of predictions from an ensemble of reward models, which is scalable and effective in *hard exploration* tasks. Lastly, extensive experiments are conducted in Robosuite, Atari games, MiniGrid, and Habitat. The results demonstrate that our algorithm can effectively enhance exploration and accelerate training across various environments.

2 Background

We focus on exploration bonuses to incentivize exploration in reinforcement learning (RL). To foster the reader's understanding, we first introduce standard notation and common practices.

Markov Decision Processes. We assume the underlying environment is a Markov decision process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s' \mid s, a)$ is state transition function from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$, R is the reward function and $\gamma \in [0,1)$ is the discount factor. Generally, the policy of an agent in an MDP is a mapping $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$. An agent chooses actions $a \in \mathcal{A}$ according to a policy function $a \sim \pi(s)$, which updates the system state $s' \sim P(s,a)$ yielding a reward $r = R(s,a)$. In this paper, we denote a policy by π_θ, where θ is the parameter of the policy function. The goal of the agent is to optimize the parameter θ for maximizing the expected accumulative rewards, $J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t\right)\right]$.

State Discrepancy as Exploration Bonuses. Addressing the sparse reward challenge prevalent in many environments necessitates augmenting the external reward function, r^e, with an intrinsic reward bonus, b, resulting in a combined reward: $r = r^e + b$. A number of intrinsic bonuses[2] that encourage exploration have been proposed. More recent methodologies employ an exploration bonus that varies based on the novelty of the current state relative to previously visited states, evaluated within different metric spaces, which can be generally defined as:

$$b = \mathtt{SD} * \mathtt{SF} = d(\Phi(s), \Phi(s')) * \lambda^{\mathtt{SF}} \tag{1}$$

where $\mathtt{SD}$ denotes the state discrepancy, $\mathtt{SF}$ denotes the scaling factor, $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ represents the evaluation metric that quantifies the degree of difference or distance between two projected states, and $\Phi : \mathcal{S} \to \mathcal{Z}$ is the projection function which maps the state to a representation space $\mathcal{Z}$.

Algorithm	Objective = State Discrepancy∗Scaling Factor	Episodic	Approximation Gap	Scalability
RIDE [51]	$\|\phi(s') - \phi(s)\|_2 * \sqrt{N_{ep}(s')}^{-1}$	✓	✗	✗
NovelD [69]	$\|r^{\mathrm{RND}}(s') - \alpha \cdot r^{\mathrm{RND}}(s)\|_+ * \mathbb{I}[N_{ep}(s') = 1]$	✓	✗	✗
LIBERTY [63]	$[\gamma d_{inv}(s', s_0) - d_{inv}(s, s_0)] * \lambda$	✗	✓	✗
EME (ours)	$d_E(s', s) * min\{max\{\zeta(r_s), 1\}, M\}$	✗	✗	✓

Table 1: Comparison of exploration methods using different measures (marked in blue) of state discrepancy as exploration bonus. $\phi = f(s)$: features encoder; $N_{ep}(s)$: episodic (pseudo) count of visits to state s; $r^{\mathrm{RND}}(s)$: the RND [11] bonus; α: normalized coefficient; d_{inv}: inverse dynamic bisimulation metric used in [63]; s_0: initial state; λ: the scaling hyper-parameter; d_E: our proposed effective metric for state discrepancy evaluation; $\zeta(r_s)$: variance of predictions from an ensemble of reward models: M: maximum reward scaling.

The comparison of recent metric-based exploration methods is summarized in Table 1 These bonuses reward high dissimilarity between adjacent states, encouraging exploration by leveraging measures such as the L_2 distance for state embeddings (as used by RIDE [51]) and the L_1 distance for the disparity in RND bonuses between adjacent states (as utilized by NovelD [69]). However, these approaches, grounded in L_p norms and dependent on episodic state visit counts, face significant scalability challenges when exploring various environments. On the other hand, LIBERTY [63] proposes the inverse dynamic bisimulation metric to evaluate the state difference, which links state and value differences to enhance learning efficiency. Yet, the approximation inaccuracies inherent in bisimulation metric learning compromise theoretical assurances, breaking theoretical integrity of bisimulation metric. In the following sections, we detail the shortcomings of these methods.

3 Limitations of Existing Metric-based Exploration Bonus

3.1 Latent Vector Space under L_p Norms

L_p norms such as L_1 and L_2 norms are widely used in measuring the "distance" between states or embeddings. For example, considering a continuous map between the states of an MDP and a latent vector space: $\Phi : \mathcal{S} \to \mathcal{Z} \subseteq \mathbb{R}^n$. We can equip this vector space with a Euclidean norm to obtain a Euclidean metric space $(\mathcal{Z}, \|\cdot\|_2)$, and the state discrepancy is measured as $\mathtt{SD} = \|\Phi(s) - \Phi(s')\|_2$.

Reliance on Count-based Scaling Factor. The state-difference based exploration bonus employed by RIDE [51] and Noveld [69] can be regarded as the difference between adjacent states in the L_p norm space. The efficacy of the exploration bonus heavily lies in the episodic scaling factor ($\mathtt{SF}$) measured by the count of visited states $N_{ep}(s)$ presented in Table 1. However, the count-based episodic bonus suffers from a fundamental limitation, which is similar to that faced by count-based approaches [5] in general: if each state is unique, then

Table 2: Mean success rate comparison (values below 0.5 are marked in red)

Environment	Robosuite	MiniGrid	Habitat
RIDE	0.52	0.93	0.19
RIDE w/o EP	0.05 ↓	0.00 ↓	0.00 ↓
NovelD	0.55	0.93	0.17
NovelD w/o EP	0.11 ↓	0.00 ↓	0.00 ↓
LIBERTY	0.73	0.00	0.13

[2] The detailed discussion of related work can be found in Appendix F

$N_{ep}(s_t)$ will always be 1 and the episodic bonus is no longer meaningful, which is the case for many real-world applications. For example, a household robot's state as recorded by its camera might include moving trees outside the window, clocks showing the time, or images on a television screen which are not relevant to its tasks, but nevertheless make each state unique. As shown in Table 2, without the episodic count term (denoted as "w/o EP"), the performance of RIDE and Noveld exhibits a significant decline in all different types of environments.

Limited Expressiveness of Novelty. As indicated in Table 2, the performance of RIDE and NovelD significantly degrades in environments with high-dimensional visual observations or continuous control tasks, where the success rate falls below 0.5. The significant decline highlights the inadequacies of utilizing state differences within vector spaces, computed using the L_p norm, to effectively express novelty. Therefore, a more robust and effective metric is essential to accurately assess and characterize the novelty among visited states.

3.2 State Discrepancy within the Bisimulation Metric Space

The bisimulation metric [25, 26] defines a pseudometric $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ to measure the similarity between two states. The recently proposed variant, π-bisimulation metric [13], focuses on behaviors relative to a particular policy π, which consists of a reward difference term and a Wasserstein distance in dynamics models between states.

Definition 1 (π-bisimulation metric)**.** *Given a fixed policy π, the following π-bisimulation metric exists and is unique:*

$$d(s_i, s_j) = |\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}| + \gamma W_1(d)(P^\pi(\cdot \mid s_i), P^\pi(\cdot \mid s_j)) \tag{2}$$

where $\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} = \mathbb{E}_{a_i \sim \pi(\cdot|s_i)} r(s_i, a_i), P^\pi(\cdot \mid s_i) = \mathbb{E}_{a \sim \pi(\cdot|s_i)} P(\cdot \mid s_i, a)$ and W_1 is the 1-Wasserstein distance.

Approximation Gap. LIBERTY [63] proposes exploration bonuses based on state discrepancy within the inverse dynamic bisimulation metric space, which adds the difference between action output by the inverse dynamic model based on the bisimulation metric. The difference between states evaluated under the bisimulation metric is directly associated with the value difference (see Theorem 3.3 in [26] and Theorem 2 in [63] for detailed proof). The loss function of LIBERTY metric learning is:

$$\mathcal{L}(\phi) = \mathbb{E}\left[\left(d_{inv}(s_i, s_j; \phi) - |r_{s_i}^{a_i} - r_{s_j}^{a_j}| - \gamma W_2(P_{\bar{s_i}}^\pi(\eta), P_{\bar{s_j}}^\pi(\eta)) - \gamma |I_{\bar{s_i}}^{s_{i+1}}(\theta) - I_{\bar{s_j}}^{s_{j+1}}(\theta)|\right)^2\right] \tag{3}$$

where $\bar{s}$ denotes state with stop gradients, $P_{\bar{s_i}}^\pi(\eta)$ indicates probabilistic dynamics model parameterized with η and $I_{\bar{s_i}}^{s_{i+1}}(\theta)$ is the inverse dynamic model parameterized with θ. As the 1-Wasserstein distance is usually difficult to estimate, LIBERTY and prior methods [68] propose to use 2-Wasserstein distance W_2 to replace W_1, as W_2 has a convenient closed-form of a Gaussian distribution with respect to the L_2 distance.

Proposition 1 (Relaxation Divergence)**.** *Relaxing the W_1 metric to W_2 breaks the theoretical integrity of the inverse dynamic bisimulation metric [63] in scenarios where the transition dynamic model $P(s, a)$ or the policy π is stochastic.*

Proof in Appendix B. Based on Proposition 1, we can see that the relaxation divergence of 1-Wasserstein distance between dynamic models can only be ignored when both the transition dynamics model and policy π are deterministic. However, this assumption may be too strong to hold in practice.

Besides, the metric learning process encounters a theory-practice gap due to the relaxation of reward expectations. Specifically, computing the expected reward differences (first term of Equation (2)), $|\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}|$, proves to be computationally daunting and challenging to accurately estimate, even with sampling techniques. Following other bisimulation metric-based methods [13, 68], LIBERTY [63] shifts the expectation operator outside the absolute value of reward differences (second term of Equation (3)) and differences between action outputs predicted by inverse dynamic models, as in $\mathbb{E}_{a_i \sim \pi, a_j \sim \pi}[|r_{s_i}^{a_i} - r_{s_j}^{a_j}| + |I_{\bar{s_i}}^{s_{i+1}} - I_{\bar{s_j}}^{s_{j+1}}|]$, which facilitates more efficient sampling by avoiding the direct estimation of reward and action expectations.

Proposition 2 (Shifted LIBERTY Distance)**.** *The relaxation of expectation during learning process shifts the original LIBERTY distance and introduces a looser value difference bound.*

We extend the concept of the shifted MICo distance [14] introduced in [35, 16]. The proof is in appendix B. Based on Proposition 2, as the shifted LIBERTY distance has a looser value difference bound, it may be less relevant to the value function. As a result, the learned metric may not be able to capture the state similarity within bisimulation metric space accurately. , as the shifted LIBERTY distance has a looser value difference bound, it may be less relevant to the value function. As a result, the learned metric may not be able to capture the state similarity within bisimulation metric space accurately.

Scalability Constraints. As shown in Table 2, LIBERTY's performance significantly drops in hard exploration tasks within MiniGrid and in the realistic scenarios of the Habitat environment [60]. Specifically, in Habitat, agents must navigate through photorealistic simulations of actual indoor spaces. In these settings, the subtle variations between states diminish the impact of the exploration bonus, significantly lowering the efficiency of exploration. This limitation becomes even more apparent in the procedurally-generated MiniGrid environment, where the inherent stochasticity further impedes LIBERTY's performance. Those findings highlight the scalability challenges that bisimulation metric-based methods encounter in complex environments.

4 Effective Metric-based Exploration Bonus

In this section, we introduce the Effective Metric-based Exploration-bonus (EME), conceived to overcome the limitations identified in existing methods that use metric-based state discrepancy as exploration bonuses. Our development is driven by two primary objectives. Firstly, we aim to refine the metric learning process to achieve a more effective exploration without incurring the approximation gap, thereby upholding the theoretical integrity of value function bound. Secondly, we seek to obviate the reliance on scaling factor of episodic counts, ensuring effective exploration across different environments. To this end, we propose a novel metric that more precisely evaluates the behavioral similarity between states, establishing a more effective exploration strategy compared with previous approaches. Furthermore, we introduce an innovative approach that utilizes the variance between outputs of an ensemble of reward models in metric learning as a dynamically-adjusted scaling factor, which significantly enhances exploration efficiency, especially in hard exploration tasks, thereby improving scalability. The next two sections describe our method in detail.

4.1 The EME Metric

In order to avoid the approximation gap introduced by relaxation of Wasserstein distance and reward expectations mentioned in Proposition 1 and 2, along with the reliance on the episodic count, we eliminate the calculation of the Wasserstein distance, and propose the EME metric:

Definition 2 (EME Distance Function). *Let $\mathfrak{met}$ be the space of bounded pseudo-metrics on state space $\mathcal{S}$, the EME metric $d_E : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, the EME distance function $\mathcal{F}(d_E, \pi) : \mathfrak{met} \to \mathfrak{met}$ is defined as:*

$$\mathcal{F}(d_E, \pi)(s_i, s_j) = |\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}| + \gamma \mathbb{E}_{\substack{a_i \sim \pi \\ a_j \sim \pi}} d_E(s_i', s_j') + \gamma D_{\mathrm{KL}}(\pi(\cdot|s_i) \| \pi(\cdot|s_j)) \quad (4)$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ represents the Kullback–Leibler (KL) divergence.

Behavioral Similarity Between States. We strictly calculate the expectation of reward difference without relaxation. The EME metric measures the distribution distance between dynamics models by computing the distance between sampled subsequent states following representation learning method [14] to avoid the computation of the Wasserstein distance. Additionally, we integrate the Kullback–Leibler (KL) divergence between policy distributions to more robustly model the "**behavioral similarity**" between states, which is more effective and scalable across diverse environments with different observations, especially in addressing the critical "Noisy-TV" problem [11] during exploration. The "Noisy-TV" problem started as a thought experiment in [11] and is commonly discussed in exploration literature [51, 52, 42]. Imagine that an RL agent is rewarded with seeking the novel experience, a TV with unpredictable random noise outputs would be able to attract the agent's attention forever. The agent obtains new rewards from state discrepancy caused by noisy TV consistently, but it fails to make any meaningful action. To counter this, we utilize differences in policy distributions across states, promoting the exploration of diverse actions rather than repetitive behavior.

Theorem 1. *The EME distance function $\mathcal{F}(d_E, \pi) : \mathfrak{met} \to \mathfrak{met}$ has a unique fixed-point $\hat{d}_E$.*

Proof in Appendix B. Theorem 1 provides a convergence guarantee for the EME distance that by iterating $\mathcal{F}(d_E, \pi)$, distance will converge to the fixed-point $\hat{d}_E$.

Theorem 2. *(Guaranteed Value difference bound) Given the EME metric d_E, states s_i and s_j, and a policy π, we have*

$$|V^\pi(s_i) - V^\pi(s_j)| \leq d_E(s_i, s_j). \tag{5}$$

Improved Exploration Efficiency. The proof can be found in Appendix B. Theorem 2 demonstrates that the EME distance between states provides an upper bound on the difference in their state values. Let's recap the definition of one-step TD error: $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$. Intuitively, if there is a large value difference between states s_t and s_{t+1}, the corresponding exploration bonus b which is calculated using $d_E(s_t, s_{t+1})$ will also be large. Consequently, the total reward for the next step, $r_{t+1} = r^e + b$ (where r^e represents the external environment reward), increases accordingly. The increase of reward leads to a larger TD error when there is a significant value difference between adjacent states, incentivizing the agent to prioritize transitions with large TD errors, which not only enhances the agent's exploration capabilities but also significantly boosts training efficiency.

4.2 Tractable Optimization of EME

Based on Equation (4), between any pair of states s_i and s_j, we can define the loss function of the EME metric learning as:

$$\mathcal{L}(\phi) = \mathbb{E}\left[\left(d_E^\phi(s_i, s_j) - |\mathbb{E}_{a_i\sim\pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j\sim\pi} r_{s_j}^{a_j}| - \gamma \mathbb{E}_{\substack{a_i\sim\pi \\ a_j\sim\pi}} d_E(s_i', s_j') - \gamma D_{\mathrm{KL}}(\pi(\cdot|s_i) \| \pi(\cdot|s_j))\right)^2\right] \tag{6}$$

where d_E^ϕ is the EME metric encoder parameterized by ϕ. The last three terms in the loss function (6) can be regarded as the regression of target metric, which approximates the difference between rewards, distances of next states and distances of policy distributions, respectively. However, the reward expectations $\mathbb{E}_{a_i\sim\pi} r_{s_i}^{a_i}$ and $\mathbb{E}_{a_j\sim\pi} r_{s_j}^{a_j}$ are computationally intractable and also difficult to estimate even based on sampling [13]. And the relaxation of reward expectations $\mathbb{E}_{\substack{a_i\sim\pi \\ a_j\sim\pi}} |r_{s_i}^{a_i} - r_{s_j}^{a_j}|$ used by previous methods [63, 68, 13] will lead to the learned metric having a looser value difference bound than the original bisimulation metric as proved in Proposition 2. Inspired by metric-based representation learning methods [35, 16], we propose to use an ensemble of reward models to approximate the reward difference more accurately.

Proposition 3. *Let r_s be a random variable over the action distribution defined by $p(r_s = r_s^a) = \pi(a|s)$, $var(r_{s_i})$ denote the variance of variable r_{s_i}, we can have:*

$$|\mathbb{E}_{a_i\sim\pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j\sim\pi} r_{s_j}^{a_j}| = \sqrt{\mathbb{E}_{\substack{a_i\sim\pi \\ a_j\sim\pi}}\left[|r_{s_i}^{a_i} - r_{s_j}^{a_j}|^2\right] - var(r_{s_i}) - var(r_{s_j})} \tag{7}$$

Proof. (Sketch) The proof is based on [14, 16] by expanding $\mathbb{E}_{\substack{a_i\sim\pi \\ a_j\sim\pi}}[|r_{s_i}^{a_i} - r_{s_j}^{a_j}|^2] - |\mathbb{E}_{a_i\sim\pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j\sim\pi} r_{s_j}^{a_j}|^2$. See detailed proof in Appendix B. □

Ensemble of Reward Models. We train an ensemble of reward models $\{g(\eta_1), \ldots, g(\eta_k)\}$ to predict the reward from the state-action pairs sampling from the buffer $\mathcal{D}_\tau$: $g(s, a; \eta) : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$ by minimizing the prediction error $\|g(s_t, a_t; \eta) - r_{t+1}\|^2$. Consequently, the variance of reward across the output of different models in the ensemble is defined as follows:

$$\mathrm{var}(r_{s_i}) \approx \zeta(r_{s_i}^{a_i}) = \mathbb{E}_{\substack{a_i\sim\pi(s_i) \\ (s_i,a_i)\sim\mathcal{D}_\tau}} \left\{\mathbb{E}_\eta\left[\|g(s_i, a_i, \eta) - \mathbb{E}_\eta[g(s_i, a_i, \eta]\|_2^2\right]\right\} \tag{8}$$

Based on Equation (6), Equation (7) and Equation (8), the tractable EME loss without approximation gap can be defined as:

$$\mathcal{L}(\phi) = \mathbb{E}_{\mathcal{D}_\tau}[(d_E^\phi(s_i, s_j) - \sqrt{|r_{s_i}^{a_i} - r_{s_j}^{a_j}|^2 - \zeta(r_{s_i}^{a_i}) - \zeta(r_{s_j}^{a_j})} - \gamma \mathbb{E}_{\substack{a_i \sim \pi \\ a_j \sim \pi}} d_E^\phi(s_i', s_j') - \gamma \mathbb{E}_{\pi(\cdot|s_i)}[\log \pi(\cdot|s_i) - \log \pi(\cdot|s_j)])^2] \tag{9}$$

Diversity-Enhanced Scaling Factor. The scaling factor of previous methods is either not scalable [51, 69, 2] or hand-crafted hyper-parameter [63], in light of this, we propose a scalable diversity-enhanced scaling factor for the intrinsic reward to further improve the efficacy of exploration. Let's recap the ensemble of reward models $\{g(\eta_1), \ldots, g(\eta_k)\}$ used in metric training (9). To maintain the diversity across the individual models $g(\eta)$, we initialize each model's parameters differently and train each of them on a subset of data randomly sampled with replacement. Each model in our ensemble is trained to predict the ground truth reward. Hence, the parts of reward obtained within the state space that have been well explored by the agent will have gathered enough data to train all models, resulting in a low variance of reward predictions from the models, when generalizing to unseen but similar parts of the reward from unvisited state-space, the areas which are novel and unexplored would still have high prediction error for all models as none of them are yet trained on such examples, resulting in higher variance of the reward prediction. Therefore, we use the variance of reward predictions as the scaling factor of intrinsic reward to encourage exploration on unvisited state space. With the maximum reward scaling M, the exploration bonus is defined as:

$$b_{t+1} = d_E(s_t, s_{t+1}) * min\{max\{\zeta(r_{s_t}), 1\}, M\} \tag{10}$$

As a result, the agent receives greater rewards when encountering novel states during training, which facilitates more effective exploration, the ablation study on scaling factor and visualization of exploration bonus is in Appendix C.5. See Algorithm 1 of Appendix D.2 for detailed description.

5 Experiments

The overall objective of our experiment is to evaluate the performance of EME in comparison to other baselines, we conduct comprehensive experiments on various settings of continuous control tasks, discrete-action hard exploration games, and real indoor environments[3] to assess the effectiveness and scalability of our algorithm. The implementation details can be found in Appendix D.

Baselines. We compare it against following competitive baseline methods. ICM [47]: a famous curiosity-driven method. RND [11]: intrinsic rewards are the prediction errors of the distillation network. E3B [32]: proposing count-based episodic bonuses under continuous state spaces. RIDE [51]: using state difference under latent space as exploration bonuses. NovelD [69]: using the state difference measured by RND bonus as intrinsic rewards. LIBERTY [63]: using the state discrepancy evaluated under the bisimulation metric space as shaping reward.

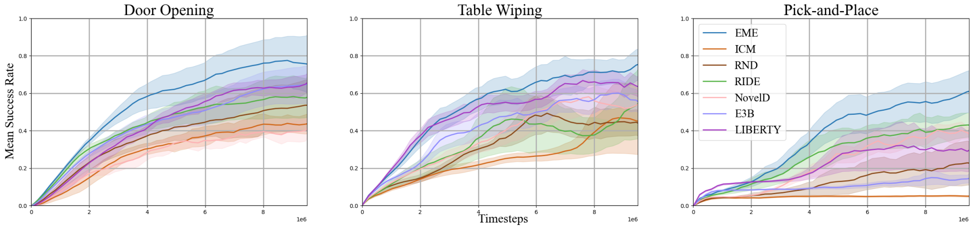


Figure 2: Results for various hard exploration tasks from Robosuite. The x-axis represents the number of steps (1e7) in training. The y-axis represents the mean success rate(standard deviations in shade).

5.1 Continuous Control

In our continuous control experiments conducted on the Robosuite platform [70], we assess the exploration capabilities of agents across various challenging tasks, which include Door Opening,

[3] Details of the environments are provided in Appendix D.1, and additional experiments are described in Appendix C.

Table 3: Average testing results of different Minigrid environments for EME and other baselines.

	MRN7S8	MRN12S10	MRN12S10-NT	KCS4R3	KCS5R3	KCS5R3-NT
ICM	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.20 ± 0.003	0.00 ± 0.0	0.00 ± 0.0
RND	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.62 ± 0.011	0.00 ± 0.0	0.00 ± 0.0
RIDE	0.67 ± 0.001	**0.65 ± 0.003**	0.60 ± 0.005	0.93 ± 0.002	0.00 ± 0.0	0.00 ± 0.0
NovelD	0.67 ± 0.001	0.65 ± 0.001	0.35 ± 0.003	**0.93 ± 0.003**	0.93 ± 0.001	0.00 ± 0.0
E3B	0.61 ± 0.003	0.50 ± 0.003	0.12 ± 0.010	0.81 ± 0.007	0.35 ± 0.002	0.00 ± 0.0
LIBERTY	0.58 ± 0.012	0.00 ± 0.0	0.00 ± 0.00	0.80 ± 0.005	0.0 ± 0.0	0.00 ± 0.0
EME	**0.69 ± 0.001**	**0.65 ± 0.003**	**0.66 ± 0.001**	0.93 ± 0.001	**0.94 ± 0.005**	**0.94 ± 0.001**
	OM2Dlh	OM2Dlhb	OM2Dlhb-NT	OM1Q	OM2Q	OM2Q-NT
ICM	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0
RND	0.95 ± 0.007	0.65 ± 0.003	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0
RIDE	0.95 ± 0.015	0.87 ± 0.016	0.35 ± 0.008	0.00 ± 0.0	0.00 ± 0.0	0.00 ± 0.0
NovelD	0.96 ± 0.005	0.89 ± 0.060	0.19 ± 0.058	0.93 ± 0.009	0.96 ± 0.060	0.12 ± 0.082
E3B	**0.97 ± 0.009**	0.89 ± 0.005	0.00 ± 0.0	0.88 ± 0.010	0.80 ± 0.025	0.00 ± 0.0
LIBERTY	0.91 ± 0.001	0.80 ± 0.001	0.00 ± 0.0	0.75 ± 0.015	0.55 ± 0.030	0.00 ± 0.0
EME	0.95 ± 0.025	**0.91 ± 0.007**	**0.90 ± 0.016**	**0.95 ± 0.012**	**0.97 ± 0.003**	**0.56 ± 0.015**

Table Wiping, and Pick-and-Place. The detailed description of environments can be found in Appendix D.1.1. Each task represents a demanding context for robotic control, characterized by sparse rewards and significant exploratory challenges. The overall results are depicted in Figure 2, where our method consistently outperforms others, demonstrating its superiority in handling continuous control tasks. LIBERTY achieves the second-best performance in the Door Opening and Table Wiping tasks, underscoring the advantages of state discrepancy-based novelty within the bisimulation metric space. Conversely, episodic count-based methods such as RIDE, NovelD, and E3B lag behind, as episodic counts become less effective in environments with high-dimensional states. Furthermore, curiosity-driven approaches like ICM and RND struggle due to insufficient exploration. We also provide additional ablation studies on the ensemble size and max reward scaling in Appendix C.

EME Combined with Feature Encoder. The EME metric can be integrated with any state representation derived from various encoders $\Phi(\cdot)$, expressed as $d_E(\Phi(s_t), \Phi(s_{t+1}))$. The encoder may include the inverse dynamic model [47, 51, 63] which isolates environmental factors that do not affect the agent's behavior, the bisimulation-based encoder [68, 35, 14, 13], and random embeddings. The results can be found in Figure 8 and 9 in Appendix C.2, the performance of EME under the inverse dynamic encoder and bisimulation-based encoder exhibits a decline, because the exploration bonus under these encoders get close to zero quickly since their representations are very compact and every state looks more similar with the convergence of the encoder, which harms the efficacy of exploration. With respect to random embedding, the variance is larger, resulting in more unstable performance.

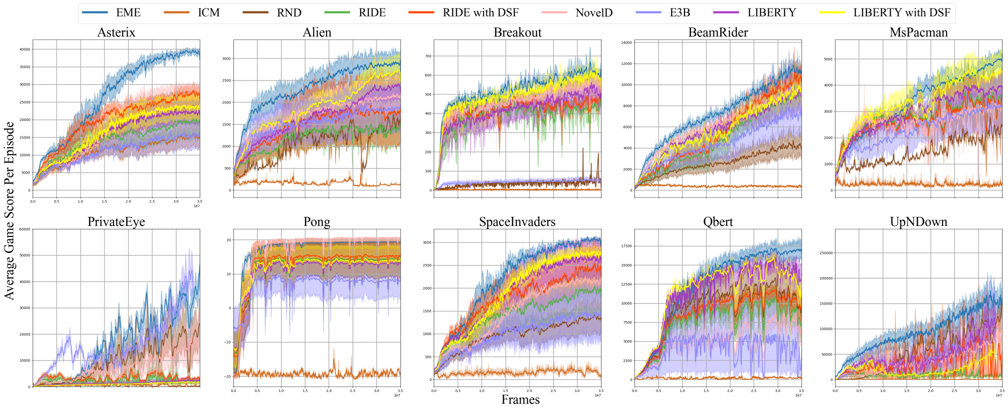


Figure 3: Comparison results for different hard exploration Atari games.

5.2 Discrete-action Games

Atari Games. To assess EME in scenarios involving pixel-based observations and discrete actions, we evaluate our method on the hard exploration games identified by [6, 69, 63] within the common

Atari benchmark [8], including Asterix, Alien, Breakout, BeamRider, Mspacman, PrivateEye, Pong, SpaceInvaders, Qbert, and UpNDown. The results are presented in Figure 3. Our method consistently achieves similar or higher average returns compared to other baselines across all tested tasks. Notably, EME demonstrates the fastest convergence, particularly in Asterix and UpNDown, highlighting our method's superior learning efficiency. While LIBERTY and NovelD exhibit competitive performance in 6 games, their effectiveness significantly diminishes in the challenging games, PrivateEye and UpNDown. As for other methods like RIDE, E3B, ICM, and RND struggle across all tasks, primarily due to inadequate exploration capabilities. To isolate the impact of our proposed EME metric, we also include the performance of RIDE and LIBERTY with Diversity-enhanced Scaling Factor, denoted as RIDE and LIBERTY with DSF. EME still achieves the best performance which demostrate the superiority of EME metric.

MiniGrid Environments. MiniGrid [17] presents a series of procedurally-generated, challenging environments. We focus on three settings from MiniGrid: Multi-Room (MR), Key Corridor (KC), and Obstructed Maze (OM). For example, MRN12S10-NT stands for MultiRoom-N12-S10-NoisyTV. Details on the specific environmental settings are available in Appendix D.1.2. Table 3 presents the testing performance of EME and all baseline methods across five different seeds. EME successfully solves all hard-level exploration environments within MiniGrid, achieving the best performance in 10 out of 12 settings. While Multi-Room environments are relatively easy, all baselines except for ICM and RND, demonstrate competency. However, as room size and number increase, the complexity of the tasks also rises, with EME consistently outperforming other baselines, a trend that extends to the Key-Corridor environments. In the particularly challenging Obstructed Maze environments, where obstructions block doors, our agent also excels by learning to remove these obstructions, further demonstrating the effectiveness and scalability of our approach.

Noisy TV Problem. In addition to standard MiniGrid tasks, we also tested the model's ability to deal with stochasticity in the environment by adding a manually-made Noisy TV setting introduced in [51], where some blocks change color at every time step. As illustrated in Table 3, EME maintains strong performance even under these conditions. Notably, EME is the only method that successfully completes the KCS5R3-NT task, underscoring the robustness of our method.

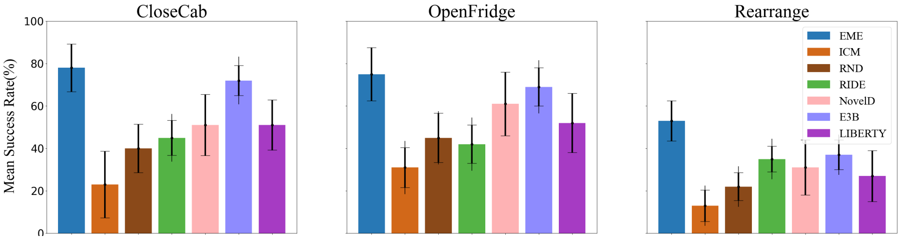


Figure 4: Results of exploration tasks on Habitat. Error bars represent std, deviations over 5 seeds.

5.3 Real-life Habitat Environment

To test our model's applicability to real-world environments, we investigate hard exploration tasks in Habitat [60]. Habitat is a platform for embodied AI research which provides an interface for agents to navigate and act in photorealistic (Figure 5) simulations of real indoor environments. Full details on the environmental setting can be found in Appendix D.1.3. We evaluate our methods on three embodied AI tasks of Habitat benchmark. As shown in Figure 4, EME consistently outperforms all baselines, demonstrating its superior scalability to high-dimensional visual-based observations and confirming its broad applicability. The performances of count-based methods like E3B, NovelD and RIDE are comparable, whereas curiosity-driven methods such as ICM and RND lag behind. The underperformance is primarily due to their inadequate exploration capabilities, which are particularly challenged by the complex and rich visual observations inherent in real-world settings.

Figure 5: Habitat

Reward-free Exploration. we investigate reward-free exploration by evaluating the performance of all baselines using exploration bonus-only. Figure 10 displays the trajectories for all baselines on one of the test maps, clearly demonstrating that EME explores a significantly larger portion of the space compared to other methods. The full experiment results can be found in Appendix C.3.

6 Conclusion

In this work, we identify the limitations of existing state discrepancy-based exploration methods: the reliance on episodic count-based scaling factor and theory-practice approximation gap, which leads to limited scalability especially for hard exploration tasks within realistic environments. To address the issues, we propose the Effective Metric-based Exploration-bonus (EME), which addresses the inherent limitations by proposing a robust metric for state discrepancy evaluation backed by comprehensive theoretical analysis. Furthermore, we propose the diversity-enhanced scaling factor integrated into the exploration bonus to enhance exploration effectiveness in particularly challenging scenarios. Extensive experiments on hard exploration tasks from continuous control, discrete-action games and realistic environments have demonstrated the effectiveness and scalability of our method.

7 Acknowledgements

This work was supported by National Science and Technology Major Project (2023ZD0121401), the Science and Technology Development Fund Macau SAR (0003/2023/RIC, 0052/2023/RIA1, 0031/2022/A, 001/2024/SKL for SKL-IOTSC), the Research Grant of University of Macau (MYRG2022-00252-FST), Shenzhen-Hong Kong-Macau Science and Technology Program Category C (SGDX20230821095159012), and Wuyi University Hong Kong and Macau joint Research Fund (2021WGALH14). This work was performed in part at SICC which is supported by SKL-IOTSC, University of Macau.

References

[1] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub W. Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Joshua Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39:20 – 3, 2018.

[2] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martin Arjovsky, Alexander Pritzel, Andrew Bolt, et al. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, 2019.

[3] Babak Badnava, Mona Esmaeili, Nasser Mozayani, and Payman Zarkesh-Ha. A new potential-based reward shaping for reinforcement learning agent. In *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 01–06. IEEE, 2023.

[4] Chenjia Bai, Peng Liu, Kaiyu Liu, Lingxiao Wang, Yingnan Zhao, Lei Han, and Zhaoran Wang. Variational dynamic for self-supervised exploration in deep reinforcement learning. *IEEE Transactions on neural networks and learning systems*, 34(8):4776–4790, 2021.

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

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

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

[8] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[9] Bernadette Bucher, Anton Arapin, Ramanan Sekar, Marc Badger, Feifei Duan, Oleh Rybkin, and Kostas Daniilidis. Perception-driven curiosity with bayesian surprise. 2019.

[10] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[11] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

[12] Andres Campero, Roberta Raileanu, Heinrich Küttler, Joshua B Tenenbaum, Tim Rocktäschel, and Edward Grefenstette. Learning with amigo: Adversarially motivated intrinsic goals. *arXiv preprint arXiv:2006.12122*, 2020.

[13] Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10069–10076, 2020.

[14] Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. *Advances in Neural Information Processing Systems*, 34:30113–30126, 2021.

[15] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[16] Jianda Chen and Sinno Pan. Learning representations via a robust behavioral metric for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:36654–36666, 2022.

[17] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

[18] C Colas, T Karch, O Sigaud, and PY Oudeyer. Intrinsically motivated goal-conditioned reinforcement learning: A short survey. arxiv 2020. *arXiv preprint arXiv:2012.09830*.

[19] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

[20] Sam Michael Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *Proceedings of the 11th international conference on autonomous agents and multiagent systems*, pages 433–440. IFAAMAS, 2012.

[21] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pages 1329–1338. PMLR, 2016.

[22] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

[23] Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. *Advances in neural information processing systems*, 32, 2019.

[24] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pages 162–169, 2004.

[25] Norman Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. *arXiv preprint arXiv:1207.4114*, 2012.

[26] Norman Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *UAI*, pages 210–219, 2014.

[27] Norman Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *UAI*, pages 210–219, 2014.

[28] Yannis Flet-Berliac, Johan Ferret, Olivier Pietquin, Philippe Preux, and Matthieu Geist. Adversarially guided actor-critic. *arXiv preprint arXiv:2102.04376*, 2021.

[29] Yang Gao and Francesca Toni. Potential based reward shaping for hierarchical reinforcement learning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[30] Marek Grzes and Daniel Kudenko. Learning potential for reward shaping in reinforcement learning with tile coding. In *Proceedings AAMAS 2008 Workshop on Adaptive and Learning Agents and Multi-Agent Systems (ALAMAS-ALAg 2008)*, pages 17–23, 2008.

[31] Anna Harutyunyan, Sam Devlin, Peter Vrancx, and Ann Nowé. Expressing arbitrary reward functions as potential-based advice. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[32] Mikael Henaff, Roberta Raileanu, Minqi Jiang, and Tim Rocktäschel. Exploration via elliptical episodic bonuses. *Advances in Neural Information Processing Systems*, 35:37631–37646, 2022.

[33] 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.

[34] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pages 4940–4950. PMLR, 2021.

[35] Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. *Advances in Neural Information Processing Systems*, 34:4764–4777, 2021.

[36] Hyoungseok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. Emi: Exploration with mutual information. *arXiv preprint arXiv:1810.01176*, 2018.

[37] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32:1238 – 1274, 2013.

[38] Weichen Li, Rati Devidze, and Sophie Fellenz. Potential-based reward shaping for learning to play text-based adventure games. *arXiv preprint arXiv:2302.10720*, 2023.

[39] Zhixuan Lin, Pierluca D'Oro, Evgenii Nikishin, and Aaron Courville. The curse of diversity in ensemble-based exploration. *arXiv preprint arXiv:2405.04342*, 2024.

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

[41] Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1706.08090*, 2017.

[42] Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Curiosity-driven exploration via latent bayesian surprise. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7752–7760, 2022.

[43] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[44] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287. Morgan Kaufmann, 1999.

[45] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[46] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

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

[48] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

[49] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, pages 835–853. PMLR, 2020.

[50] Sebastien Racaniere, Andrew K Lampinen, Adam Santoro, David P Reichert, Vlad Firoiu, and Timothy P Lillicrap. Automated curricula through setter-solver interactions. *arXiv preprint arXiv:1909.12892*, 2019.

[51] Roberta Raileanu and Tim Rocktäschel. RIDE: rewarding impact-driven exploration for procedurally-generated environments. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[52] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *arXiv preprint arXiv:1810.02274*, 2018.

[53] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. 1991.

[54] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*, pages 9443–9454. PMLR, 2021.

[55] Christopher Stanton and Jeff Clune. Curiosity search: producing generalists by encouraging individuals to continually explore and acquire skills throughout their lifetime. *PloS one*, 11(9):e0162235, 2016.

[56] Christopher Stanton and Jeff Clune. Deep curiosity search: Intra-life exploration improves performance on challenging deep reinforcement learning problems. *arXiv preprint arXiv:1806.00553*, 2018.

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

[58] Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

[59] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[60] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[61] 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.

[62] Ruo Yu Tao, Vincent François-Lavet, and Joelle Pineau. Novelty search in representational space for sample efficient exploration. *Advances in Neural Information Processing Systems*, 33:8114–8126, 2020.

[63] Yiming Wang, Ming Yang, Renzhi Dong, Binbin Sun, Furui Liu, et al. Efficient potential-based exploration in reinforcement learning using inverse dynamic bisimulation metric. *Advances in Neural Information Processing Systems*, 36, 2024.

[64] Eric Wiewiora, Garrison W Cottrell, and Charles Elkan. Principled methods for advising reinforcement learning agents. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 792–799, 2003.

[65] Peter R. Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J. Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, Leilani Gilpin, Piyush Khandelwal, Varun Kompella, HaoChih Lin, Patrick MacAlpine, Declan Oller, Takuma Seno, Craig Sherstan, Michael D. Thomure, Houmehr Aghabozorgi, Leon Barrett, Rory Douglas, Dion Whitehead, Peter Dürr, Peter Stone, Michael Spranger, and Hiroaki Kitano. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602:223 – 228, 2022.

[66] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021.

[67] Daochen Zha, Wenye Ma, Lei Yuan, Xia Hu, and Ji Liu. Rank the episodes: A simple approach for exploration in procedurally-generated environments. *arXiv preprint arXiv:2101.08152*, 2021.

[68] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

[69] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. Noveld: A simple yet effective exploration criterion. *Advances in Neural Information Processing Systems*, 34:25217–25230, 2021.

[70] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

A Broader Impact Statement

This work introduces a reinforcement learning (RL) exploration method by introducing an effective metric-based exploration bonus, a versatile framework applicable to a wide array of decision-making problems, which can be applicable to diverse scenarios, including autonomous driving, household robotics, video gaming, online recommendation or advertisement optimization and etc. Similar to other RL exploration algorithms, our method is designed to facilitate the learning of a policy that maximizes a reward function defined by the designer. The implementation of such a policy, depending on the objectives set by the reward function designer, could lead to either positive or negative outcomes. Additionally, our approach significantly reduces the burden of reward tuning for researchers and algorithm engineers, and concurrently offers insights into the quality of the designed rewards through the shaping weights derived from our methods.

B Proofs

Proposition 1 (Relaxation Divergence)**.** *Relaxing the W_1 metric to W_2 breaks the theoretical integrity of the inverse dynamic bisimulation metric [63] in scenarios where the transition dynamic model $P(s, a)$ or the policy π is stochastic.*

Proof. First, let's recall the proof of the existence of a unique fixed-point in the inverse dynamic bisimulation metric:

Lemma 1 ([63])**.** *Given a fixed policy $\hat{\pi}$. Define $\mathcal{H}^{\hat{\pi}} : \mathfrak{met} \mapsto \mathfrak{met}$ by $\mathcal{H}(d, \hat{\pi})(s_i, s_j) = |r^{\hat{\pi}}_{s_i} - r^{\hat{\pi}}_{s_j}| + \gamma W(d)(\mathcal{P}^{\hat{\pi}}_{s_i}, \mathcal{P}^{\hat{\pi}}_{s_j}) + \|I^{\hat{\pi}}(\cdot \mid s_i, s_{i+1}) - I^{\hat{\pi}}(\cdot \mid s_j, s_{j+1})\|_1$, then $\mathcal{H}^{\hat{\pi}}$ has a least fixed point $d^{\hat{\pi}}$.*

Proof. This proof mimics the proof of Theorem 4.5 from [24]. We make use of the same pointwise ordering on $\mathfrak{met}$: $d \leq d'$ iff $d(s,t) \leq d'(s,t)$ for all $s, t \in \mathcal{S}$, which gives us an ω-cpo with bottom $\perp$, which is the everywhere-zero metric. Since Lemma 4.4 from [24] (Wasserstein metric W is continuous) also applies in our definition, it only remains to show that $\mathcal{H}(d, \hat{\pi})$ is continuous:

$$\begin{aligned}
\mathcal{H}^{\hat{\pi}}\left(\bigsqcup_{n\in\mathbb{N}}\{x_n\}\right)(s_i, s_j) &= |r^{\hat{\pi}}_{s_i} - r^{\hat{\pi}}_{s_j}| + \gamma W\left(\bigsqcup_{n\in\mathbb{N}}\{x_n\}\right)(\mathcal{P}^{\hat{\pi}}_{s_i}, \mathcal{P}^{\hat{\pi}}_{s_j}) \\
&+ \|I^{\hat{\pi}}(\cdot \mid s_i, s_{i+1}) - I^{\hat{\pi}}(\cdot \mid s_j, s_{j+1})\|_1 \\
&= |r^{\hat{\pi}}_{s_i} - r^{\hat{\pi}}_{s_j}| + \gamma \sup_{n\in\mathbb{N}} W\left(x_n\right)(\mathcal{P}^{\hat{\pi}}_{s_i}, \mathcal{P}^{\hat{\pi}}_{s_j}) \\
&+ \|I^{\hat{\pi}}(\cdot \mid s_i, s_{i+1}) - I^{\hat{\pi}}(\cdot \mid s_j, s_{j+1})\|_1 \\
&\quad \text{by continuity of } W \\
&= \sup_{n\in\mathbb{N}}(|r^{\hat{\pi}}_{s_i} - r^{\hat{\pi}}_{s_j}| + \gamma W\left(x_n\right)(\mathcal{P}^{\hat{\pi}}_{s_i}, \mathcal{P}^{\hat{\pi}}_{s_j}) \\
&+ \|I^{\hat{\pi}}(\cdot \mid s_i, s_{i+1}) - I^{\hat{\pi}}(\cdot \mid s_j, s_{j+1})\|_1) \\
&= \sup_{n\in\mathbb{N}}\left\{\mathcal{H}^{\hat{\pi}}\left(x_n\right)(s_i, s_j)\right\} \\
&= \left(\bigsqcup_{n\in\mathbb{N}}\left\{\mathcal{H}^{\hat{\pi}}\left(x_n\right)\right\}\right)(s_i, s_j)
\end{aligned} \tag{11}$$

The rest of the proof follows in the same way as in [24]. □

In equation (11), the existence of a unique fixed-point in the bisimulation metric requires the continuity and monotonicity of W_1 with respect to d. The properties of continuity and monotonicity do not hold with W_2. Therefore there is no more guarantee about the fixed-point existence in LIBERTY and other approximation-based methods [68] except that both the dynamics model and the policy π are deterministic, in which case $W_2(d)$ degenerates to d [24] and Banach's fixed-point exists [35]. As for scenarios when the transition model and policy are stochastic, the theoretical guarantee of LIBERTY is broken. □

Proposition 2 (Shifted LIBERTY Distance)**.** *The relaxation of expectation during learning process shifts the original LIBERTY distance and introduces a looser value difference bound.*

Proof. The proof is based on top of [14, 16].

Lemma 2.

$$\mathbb{E}_{a_i \sim \pi, a_j \sim \pi}[|r_{s_i}^{a_i} - r_{s_j}^{a_j}| + |I_{\bar{s_i}}^{s_{i+1}} - I_{\bar{s_j}}^{s_{j+1}}|] \geq |\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}| + |\mathbb{E}_{a_i \sim \pi} I_{\bar{s_i}}^{s_{i+1}} - \mathbb{E}_{a_j \sim \pi} I_{\bar{s_j}}^{s_{j+1}}| \tag{12}$$

Proof. Since $r_{s_i}^{a_i}$ and $r_{s_j}^{a_j}$ are rewards which are scalars, we have $|r_{s_i}^{a_i} - r_{s_j}^{a_j}| \geq r_{s_i}^{a_i} - r_{s_j}^{a_j}$ and $|r_{s_j}^{a_j} - r_{s_i}^{a_i}| \geq r_{s_j}^{a_j} - r_{s_i}^{a_i}$. By taking the expectation over a_i and a_j, we can have

$$\mathbb{E}_{a_i \sim \pi, a_j \sim \pi}|r_{s_i}^{a_i} - r_{s_j}^{a_j}| \geq \mathbb{E}_{a_i \sim \pi, a_j \sim \pi}[r_{s_i}^{a_i} - r_{s_j}^{a_j}] \tag{13}$$

and

$$\mathbb{E}_{a_i \sim \pi, a_j \sim \pi}|r_{s_j}^{a_j} - r_{s_i}^{a_i}| \geq \mathbb{E}_{a_i \sim \pi, a_j \sim \pi}[r_{s_j}^{a_j} - r_{s_i}^{a_i}] \tag{14}$$

By combining Equation (13) and Equation (14), we can have:

$$\mathbb{E}_{a_i \sim \pi, a_j \sim \pi}|r_{s_i}^{a_i} - r_{s_j}^{a_j}| \geq \mathbb{E}_{a_i \sim \pi, a_j \sim \pi}[r_{s_i}^{a_i} - r_{s_j}^{a_j}] = |\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}| \tag{15}$$

Similarly, we can have:

$$\mathbb{E}_{a_i \sim \pi, a_j \sim \pi}|I_{\bar{s_i}}^{s_{i+1}} - I_{\bar{s_j}}^{s_{i+j}}| \geq \mathbb{E}_{a_i \sim \pi, a_j \sim \pi}[I_{\bar{s_i}}^{s_{i+1}} - I_{\bar{s_j}}^{s_{i+j}}] = |\mathbb{E}_{a_i \sim \pi} I_{\bar{s_i}}^{s_{i+1}} - \mathbb{E}_{a_j \sim \pi} I_{\bar{s_j}}^{s_{j+1}}| \tag{16}$$

So we can get

$$\mathbb{E}_{a_i \sim \pi, a_j \sim \pi}[|r_{s_i}^{a_i} - r_{s_j}^{a_j}| + |I_{\bar{s_i}}^{s_{i+1}} - I_{\bar{s_j}}^{s_{j+1}}|] \geq |\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}| + |\mathbb{E}_{a_i \sim \pi} I_{\bar{s_i}}^{s_{i+1}} - \mathbb{E}_{a_j \sim \pi} I_{\bar{s_j}}^{s_{j+1}}| \tag{17}$$

□

Definition 3 (Shifted LIBERTY Distance)**.** *The shifted LIBERTY distance function $\hat{\mathcal{H}}^\pi$ is defined as*

$$\hat{\mathcal{H}}^\pi(d)(s_i, s_j) = \mathbb{E}_{\substack{a_i \sim \pi \\ a_j \sim \pi}}|r_{s_i}^{a_i} - r_{s_j}^{a_j}| - \gamma W_1(P_{\bar{s_i}}^\pi, P_{\bar{s_j}}^\pi) - \gamma|I_{\bar{s_i}}^{s_{i+1}} - I_{\bar{s_j}}^{s_{j+1}}| \tag{18}$$

Define the MDP for RL by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$. We consider a lifted MDP constructed by a tuple $\langle \hat{\mathcal{S}}, \hat{\mathcal{A}}, \hat{\mathcal{R}}, \hat{\mathcal{P}}, \gamma \rangle$, where state space $\hat{\mathcal{S}} = \mathcal{S} \times \mathcal{S}$, action space $\hat{\mathcal{A}} = \mathcal{A} \times \mathcal{A}$, transition distribution $\hat{\mathcal{P}}_{(s_i, s_j)}^{(a_i, a_j)} = P_{s_i}^{a_i} P_{s_j}^{a_j}$, and reward function $\hat{\mathcal{R}}((s_i, s_j)) = |\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}|$. The Bellman operator T^π under policy $\pi(a_i, a_j | s_i, s_j) = \pi(a_i | s_i)\pi(a_j | s_j)$ is:

$$\begin{aligned} T^\pi(d^\pi)((s_i, s_j)) &= \sum_{(a_i, a_j)} \pi(a_i, a_j | s_i, s_j) \sum_{(s_i', s_j')} \hat{\mathcal{P}}_{(s_i, s_j)}^{(a_i, a_j)}(s_i', s_j') \left[\hat{\mathcal{R}}((s_i, s_j)) + \gamma |I_{s_i}^{s_i'} - I_{s_j}^{s_j'}| \right] \\ &= \left| \mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} \right| + \gamma W(P_{\bar{s_i}}^\pi, P_{\bar{s_j}}^\pi) + \gamma |I_{s_i}^{s_i'} - I_{s_j}^{s_j'}| \\ &= \mathcal{H}^\pi(d^\pi)((s_i, s_j)). \end{aligned} \tag{19}$$

So we can construct the lifted MDP of shift LIBERTY by a tuple $\langle \hat{\mathcal{S}}, \hat{\mathcal{A}}, \tilde{\mathcal{R}}, \hat{\mathcal{P}}, \gamma \rangle$ which is the same as the lifted MDP in Definition 3 except for the reward function $\tilde{\mathcal{R}}((s_i, s_j), (a_i, a_j)) = |r_{s_i}^{a_i} - r_{s_j}^{a_j}|$ and shifted metric $\tilde{d}$. The Bellman operator $\tilde{T}^{\hat{\pi}}$ under policy $\hat{\pi}$ is:

$$\begin{aligned} \tilde{T}^{\hat{\pi}}(\tilde{d}^{\hat{\pi}})((s_i, s_j)) &= \sum_{(a_i, a_j)} \hat{\pi}(a_i, a_j | s_i, s_j) \sum_{(s_i', s_j')} \hat{\mathcal{P}}_{(s_i, s_j)}^{(a_i, a_j)}(s_i', s_j') \left[\tilde{\mathcal{R}}((s_i, s_j), (a_i, a_j)) + \gamma |I_{s_i}^{s_i'} - I_{s_j}^{s_j'}| \right] \\ &= \mathcal{H}^{\hat{\pi}}(\tilde{d}^{\hat{\pi}})((s_i, s_j)) \end{aligned} \tag{20}$$

As proven in LIBERTY [63], for the inverse dynamic bisimulation metric d, we can get

$$|V^\pi(s_i) - V^\pi(s_j)| \leq d^\pi(s_i, s_j) \tag{21}$$

As proved in [63, 24, 27], d^π is the value function of lifted MDP $\langle \hat{\mathcal{S}}, \hat{\mathcal{A}}, \hat{\mathcal{R}}, \hat{\mathcal{P}}, \gamma \rangle$. d^π can be expanded as the sum of discounted future rewards,

$$d^\pi(s_i, s_j) = \mathbb{E}_{\hat{\pi}} \left[\sum_t \gamma^t \left(\mathbb{E}_{a_i^{(t)} \sim \pi} r_{s_i^{(t)}}^{a_i^{(t)}} - \mathbb{E}_{a_j^{(t)} \sim \pi} r_{s_j^{(t)}}^{a_j^{(t)}} \right) \middle| s_i^{(0)} = s_i, s_j^{(0)} = s_j \right]. \tag{22}$$

$\tilde{d}^\pi$ can be expanded as,

$$\tilde{d}^\pi(s_i, s_j) = \mathbb{E}_{\hat{\pi}} \left[\sum_t \gamma^t \left(\left| \mathbb{E}_{a_i^{(t)} \sim \pi} r_{s_i^{(t)}}^{a_i^{(t)}} - \mathbb{E}_{a_j^{(t)} \sim \pi} r_{s_j^{(t)}}^{a_j^{(t)}} \right| \right) \middle| s_i^{(0)} = s_i, s_j^{(0)} = s_j \right]. \tag{23}$$

with $s_i^{(0)} = s_i, s_j^{(0)} = s_j$, The difference between d and $\tilde{d}$ can be regarded as

$$\begin{aligned} \tilde{d}^\pi(s_i, s_j) - d^\pi(s_i, s_j) &= \mathbb{E}_{\hat{\pi}} \left[\sum_t \gamma^t \left(\left| \mathbb{E}_{a_i^{(t)} \sim \pi} r_{s_i^{(t)}}^{a_i^{(t)}} - \mathbb{E}_{a_j^{(t)} \sim \pi} r_{s_j^{(t)}}^{a_j^{(t)}} \right| - \left| \mathbb{E}_{a_i^{(t)} \sim \pi} r_{s_i^{(t)}}^{a_i^{(t)}} - \mathbb{E}_{a_j^{(t)} \sim \pi} r_{s_j^{(t)}}^{a_j^{(t)}} \right| \right) \right] \\ &= \mathbb{E}_{\hat{\pi}} \left[\sum_t \gamma^t \hat{\mathcal{R}}_\Delta((s_i^{(t)}, s_j^{(t)}), (a_i^{(t)}, a_j^{(t)})) \right] \end{aligned} \tag{24}$$

As proved in Lemma 2, we can get:

$$\tilde{d}^\pi(s_i, s_j) - d^\pi(s_i, s_j) \geq 0 \tag{25}$$

Combined with Equation (21), we can have:

$$|V^\pi(s_i) - V^\pi(s_j)| \leq d^\pi(s_i, s_j) \leq \tilde{d}^\pi(s_i, s_j) \tag{26}$$

So the shifted LIBERTY distance introduces a looser value bound. □

Proposition 3. *Let r_s be a random variable over the action distribution defined by $p(r_s = r_s^a) = \pi(a|s)$, var$(r_{s_i})$ denote the variance of variable r_{s_i}, we can have:*

$$|\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}| = \sqrt{\mathbb{E}_{\substack{a_i \sim \pi \\ a_j \sim \pi}} \left[\left| r_{s_i}^{a_i} - r_{s_j}^{a_j} \right|^2 \right] - \mathit{var}(r_{s_i}) - \mathit{var}(r_{s_j})} \tag{27}$$

Proof. The proof is based on [14, 16] by expanding the difference between $\mathbb{E}_{a_i \sim \pi, a_j \sim \pi} \left[\left(r_{s_i}^{a_i} - r_{s_j}^{a_j} \right)^2 \right]$ and $\left(\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} \right)^2$.

$$\begin{aligned} &\mathbb{E}_{a_i \sim \pi, a_j \sim \pi} \left[\left(r_{s_i}^{a_i} - r_{s_j}^{a_j} \right)^2 \right] - \left(\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} \right)^2 \\ &= \mathbb{E}_{a_i \sim \pi} \left[\left(r_{s_i}^{a_i} \right)^2 \right] + \mathbb{E}_{a_j \sim \pi} \left[\left(r_{s_j}^{a_j} \right)^2 \right] - 2\mathbb{E}_{a_i \sim \pi, a_j \sim \pi} \left[r_{s_i}^{a_i} r_{s_j}^{a_j} \right] \\ &\quad - \left(\left(\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} \right)^2 + \left(\mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} \right)^2 - 2\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} \right) \\ &= \mathbb{E}_{a_i \sim \pi} \left[\left(r_{s_i}^{a_i} \right)^2 \right] - \left(\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} \right)^2 + \mathbb{E}_{a_j \sim \pi} \left[\left(r_{s_j}^{a_j} \right)^2 \right] - \left(\mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} \right)^2 \\ &\quad - 2 \left(\mathbb{E}_{a_i \sim \pi, a_j \sim \pi} \left[r_{s_i}^{a_i} r_{s_j}^{a_j} \right] - \mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} \right) \\ &= \mathrm{var}[r_{s_i}] + \mathrm{var}(r_{s_j}) - 2\mathrm{cov}(r_{s_i}, r_{s_j}) \end{aligned} \tag{28}$$

r_{s_i} and r_{s_j} are independent variables, we can get $\mathrm{cov}(r_{s_i}, r_{s_j}) = 0$, so we can have

$$|\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}| = \sqrt{\mathbb{E}_{\substack{a_i \sim \pi \\ a_j \sim \pi}} \left[\left| r_{s_i}^{a_i} - r_{s_j}^{a_j} \right|^2 \right] - \mathrm{var}(r_{s_i}) - \mathrm{var}(r_{s_j})} \tag{29}$$

□

Theorem 1. *The EME distance function $\mathcal{F}(d_E, \pi) : \mathfrak{met} \to \mathfrak{met}$ has a unique fixed-point $\hat{d}_E$.*

Proof. Let $d_E, d_E' \in \mathbb{M}$. We have

$$\begin{aligned} \left|\mathcal{F}(d_E)\left(s_i, s_j\right) - \mathcal{F}\left(d_E'\right)\left(s_i, s_j\right)\right| &= \left|\gamma \sum_{a_i, a_j} \pi\left(a_i \mid s_i\right) \pi\left(a_j \mid s_j\right)\left(d_E - d_E'\right)\left(s_i', s_j'\right)\right| \\ &\leq \gamma \left\|d_E - d_E'\right\|_\infty \end{aligned} \tag{30}$$

Therefore, $\mathcal{F}$ is a contraction mapping w.r.t. the L_∞ norm and there exists a unique fixed-point for $\mathcal{F}$ due to Banach's fixed-point theorem. This completes the proof. □

Theorem 2. *(Guaranteed Value difference bound) Given the EME metric d_E, states s_i and s_j, and a policy π, we have*

$$\left|V^\pi\left(s_i\right) - V^\pi\left(s_j\right)\right| \leq d_E\left(s_i, s_j\right). \tag{31}$$

Proof. The proof mimics [14]. We follow the assumption that $\sum_{s'} P_s^a(s')V^\pi(s') = V^\pi(\mathbb{E}_{s' \sim P_s^a}[s'])$. We will first show that if $\forall s_i, s_j \in \mathcal{S}, |V^\pi(s_i) - V^\pi(s_j)| \leq d(s_i, s_j)$, then $|V^\pi(s_i) - V^\pi(s_j)| \leq \mathcal{F}(d_E, \pi)(s_i, s_j)$. Suppose $|V^\pi(s_i) - V^\pi(s_j)| \leq d(s_i, s_j)$ holds, we can have:

$$\begin{aligned} & V^\pi(s_i) - V^\pi(s_j) \\ & = \mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j} + \sum_{a_i} \pi(a_i|s_i) \sum_{s_i'} P_{s_i}^{a_i}(s_i')V^\pi(s_i') - \sum_{a_j} \pi(a_j|s_j) \sum_{s_j'} P_{s_j}^{a_j}(s_j')V^\pi(s_j') \\ & \leq \left|\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}\right| + \mathbb{E}_{a_i \sim \pi}(\sum_{s_i'} P_{s_i}^{a_i}(s_i')V^\pi(s_i')) - \mathbb{E}_{a_j \sim \pi}(\sum_{s_j'} P_{s_j}^{a_j}(s_j')V^\pi(s_j')) \\ & \leq \left|\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}\right| + \mathbb{E}_{a_i \sim \pi}\left(V^\pi\left(\mathbb{E}_{s_i' \sim P_{s_i}^{a_i}}[s_i']\right) - V^\pi\left(\mathbb{E}_{s_j' \sim P_{s_j}^{a_j}}[s_j']\right)\right) \\ & \leq \left|\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}\right| + \mathbb{E}_{a_i \sim \pi} d\left(\mathbb{E}_{s_i' \sim P_{s_i}^{a_i}}[s_i'], \mathbb{E}_{s_j' \sim P_{s_j}^{a_j}}[s_j']\right) \\ & \leq \left|\mathbb{E}_{a_i \sim \pi} r_{s_i}^{a_i} - \mathbb{E}_{a_j \sim \pi} r_{s_j}^{a_j}\right| + \gamma \mathbb{E}_{a_i \sim \pi} d\left(s_i', s_j'\right) + \gamma \mathbb{E}_{\pi(\cdot|s_i)}[\log \pi(\cdot|s_i) - \log \pi(\cdot|s_j)] \\ & \leq \mathcal{F}(d, \pi)(s_i, s_j)) \end{aligned} \tag{32}$$

Similarly, $V^\pi(s_j) - V^\pi(s_i) \leq \mathcal{F}(d, \pi(s_j, s_i))$, so we can have

$$|V^\pi(s_i) - V^\pi(s_j)| \leq \mathcal{F}(d, \pi)(s_i, s_j)) \tag{33}$$

Assuming that we have an initial distance d_0 which $|V^\pi(s_i) - V^\pi(s_j)| \leq d_0(s_i, s_j)$, and base on Theorem 1, $\mathcal{F}(d, \pi)$ is contraction mapping on d. By repeatedly applying $\mathcal{F}(d, \pi)$ on d, d will eventually converge to the fixed-point d_E, the fixed point d_E satisfies:

$$|V^\pi(s_i) - V^\pi(s_j)| \leq d_E(s_i, s_j)) \tag{34}$$

□

C Additional Experiments

C.1 Additional Ablation Experiments

To isolate the impact of different hyper-parameters, we carry additional experiments on the ensemble size of reward models and the max reward scaling M. First, M sets an upper limit on the bonus. We set $M = 10$as the default setting. As we can see from Figure 6, if $M = 1$, the scaling factor is fixed to 1, resulting in a significant performance decline. Higher M encourages more extensive exploration. The performance with $M = 5$ slightly lags behind the default setting. The performance with $M = 20$ and $M = 40$ is comparable and almost the same, indicating that the performance stabilizes as M increases. Practically, the value of M can be adjusted depending on the specific task and environment to determine the intensity of exploration.

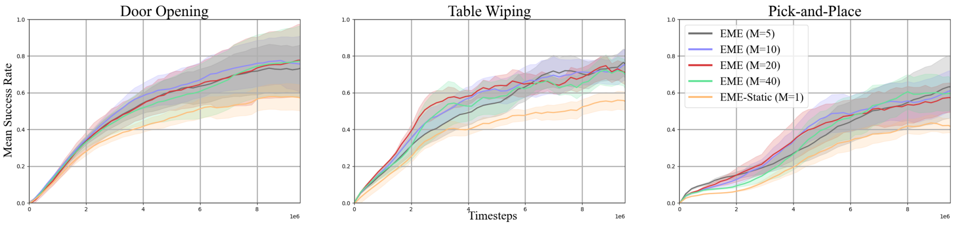


Figure 6: Ablation study on the max reward scaling M ($M = 1, 5, 10, 20, 40$).

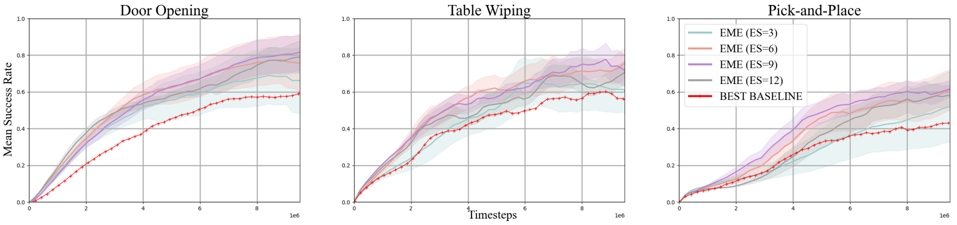


Figure 7: Ablation study on the ensemble size (ES = 3, 6, 9, 12) and the other baseline with best performance.

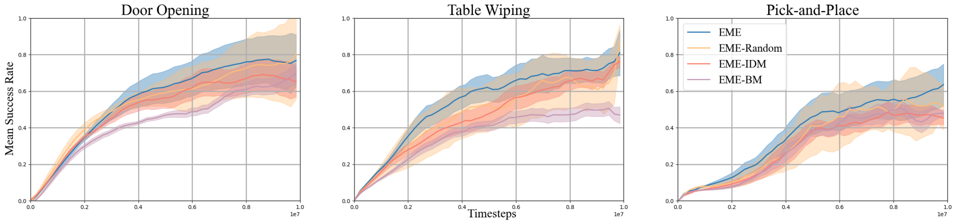


Figure 8: Results for EME and its variants combined with different feature encoders. The x-axis represents the number of steps (1e7) in training. The y-axis represents the mean success rate(standard deviations in shade).

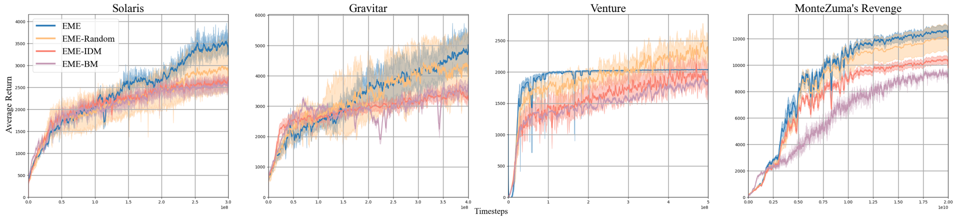
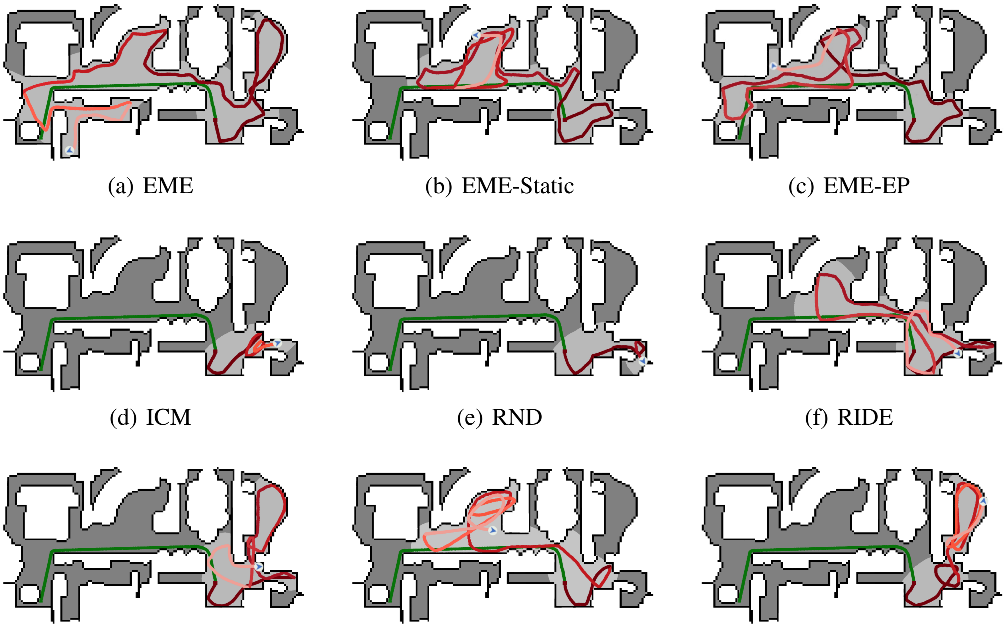


Figure 9: Results for EME and its variants combined with different feature encoders on more Atari games.

As shown in Figure 7, we observe that as the ensemble size (ES) increases, the performance of ES = 6 and ES = 9 surpasses that of ES = 3. The performance of ES = 6 and ES = 9 is comparable, with no significant difference. However, when the size is further increased to ES = 12, there is a decline in performance, particularly in the Table Wiping and Pick and Place tasks. The performance change can be analyzed from two perspectives. First, the number of reward models is related to the accuracy of the reward variance prediction used to calculate the loss function. As the ensemble size increases, the overall prediction error decreases because the models can average out individual errors more effectively. This leads to a more accurate approximation of the variance $\zeta(s_t)$. Second, with a larger ensemble size, the variance of reward predictions decreases, resulting in a lower scale of the exploration bonus. Thus, there is a balance or trade-off between estimation accuracy and exploration, explaining why the performance of ES = 12 lags behind ES = 9. The optimal number of ensemble models may vary depending on the specific tasks and environments. Regarding computational cost, it increases with the ensemble size. Therefore, we set ES = 6 as our default setting, where the performance is nearly the same as ES = 9 but with lower computational overhead. It is also noteworthy that even with an ensemble size of 3, EME still outperforms the best baseline methods, further demonstrating the robustness of EME.

C.2 EME Combined with Feature Encoder

Our method EME can be integrated with any state representation derived from different state encoders $\Phi(\cdot)$, expressed as $d_E(\Phi(s_t), \Phi(s_{t+1}))$. The encoder may include the inverse dynamic model [47, 51, 63] which isolates environmental factors that do not affect the agent's behavior, the bisimulation-based encoder [68, 35, 14, 13] which learns a compact state representation grouped by the bisimulation metric, and random embeddings captured by random embedded states. We denote the EME variants as follows: EME with an inverse dynamic model-based encoder is referred to as EME-IDM; with a bisimulation metric-based encoder as EME-BM; and with a random embedding as EME-Random. The results of continuous control tasks and hard exploration atari games can be found in Figure 8 and Figure 9, respectively. EME's performance declines due to the rapid reduction of the exploration bonus. The performance of EME under the inverse dynamic encoder and bisimulation-based encoder exhibits a decline, because the exploration bonus under these encoders gets close to zero quickly since their representations are very compact and every state looks more similar with the convergence of the encoder, which harms the efficacy of exploration. With respect to random embedding, the variance is larger, resulting in more unstable performance.



(a) EME (b) EME-Static (c) EME-EP

(d) ICM (e) RND (f) RIDE

(g) NovelD (h) E3B (i) LIBERTY

Figure 10: Trajectories of policies trained with different exploration algorithms on the Habitat environment. Our method EME reveals the largest portion of the map than other methods.

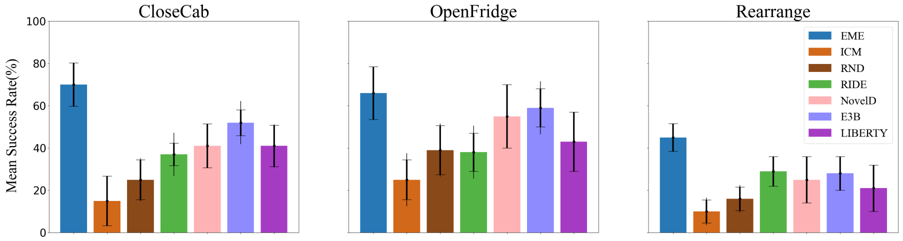


Figure 11: Results of reward-free exploration tasks on Habitat. Error bars represent std, deviations over 5 seeds.

C.3 Reward-free Exploration

We examine reward-free exploration by assessing the performance of all baselines within Habitat using only exploration bonuses. During training, agents are initialized in varied environments for each episode and are subsequently tested on a set of environments not encountered during training. Figure 10 displays the trajectories of all baselines on one of the test maps, highlighting that EME explores a substantial portion of the space, unlike other methods. Specifically, ICM and RND agents remain confined to a single room, while RIDE, LIBERTY, and NovelD agents explore two rooms before becoming stuck, distant from the goal. E3B exhibits the second-best exploration coverage of the map. Quantitative results from other reward-free tasks are shown in Figure 11, where EME consistently demonstrates robust performance and achieves the best outcomes. In contrast, other methods experience significant declines, underscoring the efficacy of our approach in a reward-free setting.

Table 4: Table of quantitative results comparison between EME and other baseline methods in different environments of Mujoco with the delayed reward setting. The best and the runner-up results are (**bold**) and ($\underline{\text{underline}}$)

Methods	Delay = 10					
	HalfCheetah	Hopper	Walker2d	Ant	Humanoid	Swimmer
ICM	1374 ± 368	1258 ± 325	1127 ± 225	-105 ± 43	462 ± 54	27 ± 11
RND	1694 ± 495	1976 ± 458	1405 ± 262	143 ± 17	532 ± 29	32 ± 15
RIDE	2467 ± 456	1876 ± 431	1651 ± 325	92 ± 31	570 ± 45	65 ± 16
NovelD	1785 ± 423	649 ± 145	975 ± 360	-134 ± 27	258 ± 49	13 ± 11
E3B	1025 ± 347	1474 ± 129	1997 ± 115	66 ± 27	518 ± 23	43 ± 17
LIBERTY	$\mathbf{2973} \pm \mathbf{437}$	$\mathbf{2479} \pm \mathbf{315}$	$\underline{2766 \pm 487}$	$\underline{292 \pm 68}$	$\mathbf{681} \pm \mathbf{73}$	$\mathbf{73} \pm \mathbf{21}$
EME	$\underline{2779 \pm 412}$	$\underline{2369 \pm 314}$	$\mathbf{2785} \pm \mathbf{399}$	$\mathbf{539} \pm \mathbf{62}$	$\underline{599 \pm 65}$	$\mathbf{77} \pm \mathbf{18}$
Methods	Delay = 20					
	HalfCheetah	Hopper	Walker2d	Ant	Humanoid	Swimmer
ICM	1185 ± 287	1097 ± 275	995 ± 201	-175 ± 23	434 ± 48	23 ± 10
RND	1595 ± 415	1925 ± 401	1379 ± 193	$127 + 12$	$\underline{519 \pm 25}$	29 ± 12
RIDE	2285 ± 402	1621 ± 382	$\underline{1724 \pm 307}$	105 ± 27	509 ± 21	59 ± 13
NovelD	945 ± 355	513 ± 86	794 ± 320	-107 ± 29	315 ± 55	17 ± 11
E3B	887 ± 242	1015 ± 185	743 ± 95	381 ± 45	321 ± 36	20 ± 11
LIBERTY	$\mathbf{2619} \pm \mathbf{354}$	$\underline{2112 \pm 208}$	$\mathbf{2345} \pm \mathbf{414}$	$\underline{263 \pm 55}$	$\mathbf{617} \pm \ \mathbf{53}$	$\mathbf{67} \pm \mathbf{18}$
EME	$\underline{2607 \pm 276}$	$\mathbf{2467} \pm \mathbf{298}$	1718 ± 163	$\mathbf{352} \pm \mathbf{35}$	499 ± 62	$\underline{59 \pm 15}$
Methods	Delay = 30					
	HalfCheetah	Hopper	Walker2d	Ant	Humanoid	Swimmer
ICM	1017 ± 276	965 ± 213	798 ± 199	-198 ± 25	417 ± 45	19 ± 11
RND	1483 ± 393	1773 ± 391	1038 ± 191	$99 + 13$	$\underline{501 \pm 27}$	24 ± 11
RIDE	$\underline{1973 \pm 369}$	1405 ± 315	1345 ± 305	87 ± 21	487 ± 25	$\underline{41 \pm 15}$
NovelD	885 ± 217	505 ± 148	664 ± 129	-205 ± 43	279 ± 65	18 ± 10
E3B	997 ± 159	1246 ± 101	1007 ± 142	$\underline{221 \pm 25}$	366 ± 26	19 ± 12
LIBERTY	$\underline{273 \pm 317}$	$\underline{1873 \pm 228}$	$\mathbf{2077} \pm \mathbf{398}$	215 ± 48	$\mathbf{587} \pm \mathbf{63}$	$\underline{52 \pm 15}$
EME	$\mathbf{2448} \pm \mathbf{263}$	$\mathbf{1999} \pm \mathbf{288}$	$\underline{1685 \pm 158}$	$\mathbf{315} \pm \mathbf{29}$	495 ± 36	$\mathbf{58} \pm \mathbf{13}$

C.4 Delayed Reward Setting of Mujoco

We present results from the delayed reward setting in the MuJoCo environment [21], following the methodology outlined in [63], where accumulated rewards are delayed every 10, 20, and 30 steps. The experiments assess six tasks: HalfCheetah, Hopper, Walker2d, Ant, Swimmer, and Humanoid. As shown in Table 4, our method achieves the best or second-best performance in 15 out of 18 delayed reward tasks. This indicates that EME can facilitate effective exploration and maintain high performance even under sparse reward conditions. Notably, as rewards become sparser, EME's performance improves and becomes more robust. In contrast, other metric-based exploration bonus methods such as RIDE and NovelD struggle due to their less expressive metrics. Curiosity-driven methods like ICM and RND also become less effective as rewards grow sparser. These results provide further evidence of EME's ability to promote efficient exploration effectively, even in environments characterized by delayed rewards.

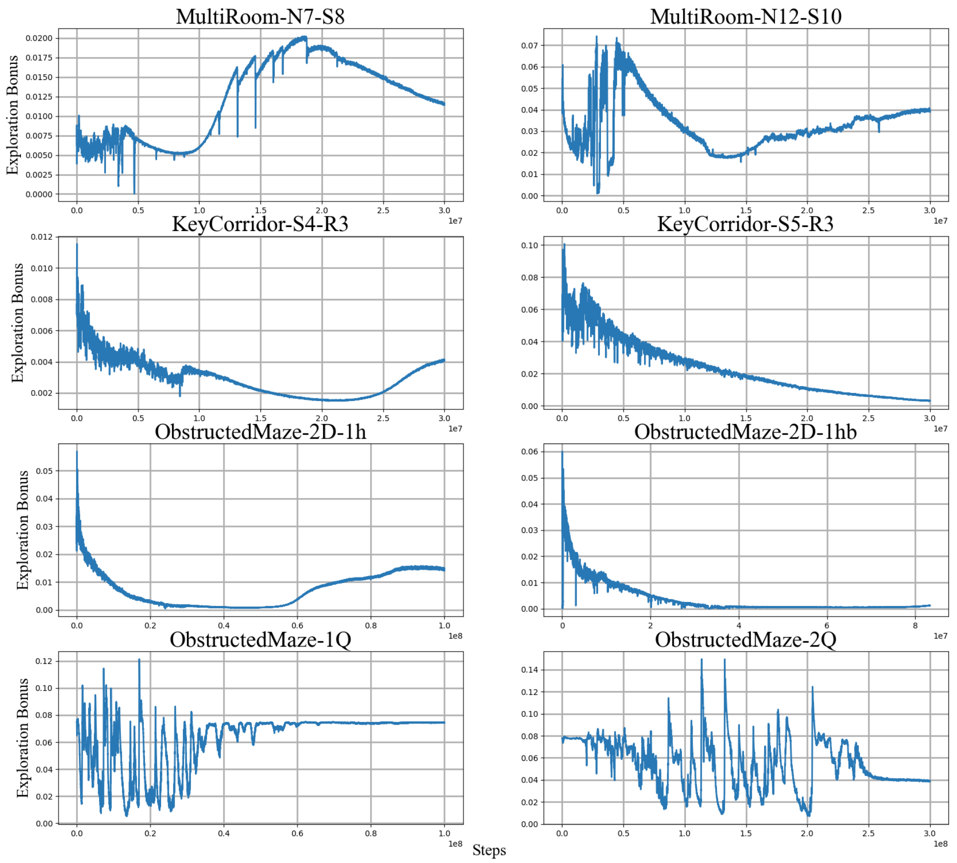


Figure 12: The scale of exploration bonus across all MiniGrid environments

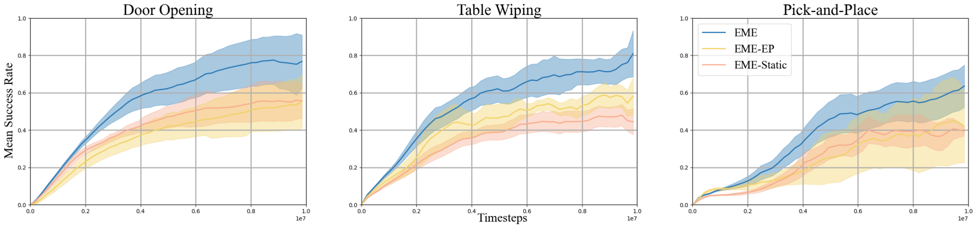


Figure 13: Results of EME, EME with a static scaling factor, and EME with an episodic count from Robosuite. The x-axis represents the number of steps (1e7) in training. The y-axis represents the mean success rate(standard deviations in shade).

Table 5: Average testing results of different Minigrid environments for EME and its variants.

	MRN7S8	MRN12S10	MRN12S10-NT	KCS4R3	KCS5R3	KCS5R3-NT
EME	0.69 ± 0.001	0.65 ± 0.003	0.66 ± 0.001	0.93 ± 0.001	0.94 ± 0.005	0.94 ± 0.001
EME-EP	0.66 ± 0.003	0.55 ± 0.011	0.20 ± 0.003	0.91 ± 0.001	0.90 ± 0.011	0.50 ± 0.009
EME-Static	0.66 ± 0.001	0.41 ± 0.005	0.00 ± 0.0	0.88 ± 0.012	0.71 ± 0.016	0.11 ± 0.003

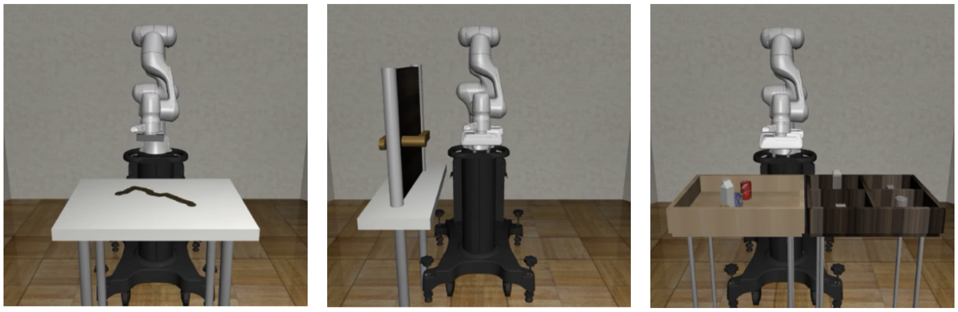
	OM2Dlh	OM2Dlhb	OM2Dlhb-NT	OM1Q	OM2Q	OM2Q-NT
EME	0.95 ± 0.025	0.91 ± 0.007	0.90 ± 0.016	0.95 ± 0.012	0.97 ± 0.003	0.56 ± 0.015
EME-EP	0.95 ± 0.016	0.89 ± 0.013	0.89 ± 0.007	0.93 ± 0.015	0.91 ± 0.023	0.20 ± 0.005
EME-Static	0.91 ± 0.008	0.80 ± 0.009	0.90 ± 0.016	0.87 ± 0.043	0.50 ± 0.033	0.00 ± 0.0

C.5 Ablation Study on Scaling Factor

To demonstrate the effectiveness of our proposed scaling factor, we conducted an ablation study focusing on this component. We compared EME with two variants: 'EME-EP,' which incorporates episodic counts, and ’EME-Static,’ which uses a static scaling factor. The results from continuous tasks in the Robosuite and MiniGrid environments are illustrated in Figure 13, Figure 10 and summarized in Table 5, respectively. Without the diversity-enhanced scaling factor, EME shows a noticeable decline in performance. ’EME-EP’ performs better in MiniGrid environments compared to continuous tasks in Robosuite, highlighting the efficacy of episodic state visitation counts in grid-based games but demonstrating limited effectiveness in environments with high-dimensional states. Additionally, we visualize the exploration bonus of all MiniGrid environments in Figure 12.

D Implementation Details

D.1 Environments Settings



(a) Table Wiping (b) Door Opening (c) Pick and Place

D.1.1 Robosuite Manipulation Tasks

We choose the three manipulation tasks in the Robosuite platform [70].

Table Wiping. The environment consists of a table with a whiteboard surface and some markings is placed in front of a single robot arm, which has a whiteboard eraser mounted on its hand. The shape of the dirty region is correlated to the position of the cube. When the dirty region is diagonal, the cube is on the right-hand side of the robot arm. The goal of the agent is to learn to wipe the whiteboard surface and clean all of the markings. The whiteboard markings are randomized at the beginning of each episode.

Door Opening. The environment consists of a door with a handle mounted in free space in front of a single robot arm. The height of the handle and the position of the door are correlated. When the door is closed to the robot arm, the handle is in a low position. When the door is far from the robot arm, the handle is in a high position. The goal of the agent is to learn to turn the handle and open the door. The initial state distribution of the door location is randomized at the beginning of each episode.

Pick and Place. The environment consists of four objects placed in a bin in front of a single robot arm. There are four containers next to the bin. The goal of the agent is to place each object into its corresponding container. This task also has easier single-object variants. The initial state distribution of object locations is randomized at the beginning of each episode.

D.1.2 MiniGrid Environments

MiniGrid [17] is a set of procedurally-generated grid-worlds. In MiniGrid, the world is a partially observable grid of size $N \times N$. Each tile in the grid contains exactly zero or one object. The possible object types are wall, door, key, ball, box, and goal. Each object in MiniGrid has an associated discrete color, which can be one of red, green, blue, purple, yellow, or grey. By default, walls are always grey and goal squares are always green. Rewards are sparse for all MiniGrid environments. There are seven actions in MiniGrid: turn left, turn right, move forward, pick up an object, drop an object, toggle and done. The agent can use the turn left and turn right action to rotate and face one of 4 possible directions (north, south, east, west).

The move forward action makes the agent move from its current tile onto the tile in the direction it is currently facing, provided there is nothing on that tile, or that the tile contains an open door. The agent can open doors if they are right in front of it by using the toggle action. Observations in MiniGrid are partial and egocentric. By default, the agent sees a square of 7 × 7 tiles in the direction it is facing. These include the tile the agent is standing on. The agent cannot see through walls or closed doors. The observations are provided as a tensor of shape 7 × 7 × 3. However, note that these are not RGB images. Each tile is encoded using 3 integer values: one describing the type of object contained in the cell, one describing its color, and a flag indicating whether doors are open or closed. This compact encoding was chosen for space efficiency and to enable faster training. For all tasks, the agent gets an egocentric view of its surroundings, consisting of 3 × 3 pixels. A neural network parameterized as a CNN is used to process the visual observation.

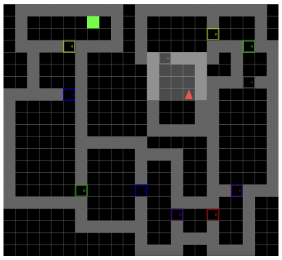


Figure 14: Rendering of the MultiRoomN12S10 in MiniGrid.

MultiRoom. The MultiRoomNXSY environment consists of X rooms, with a size at most Y, connected in random orientations. The agent is placed in the first room and must navigate to a green goal square in the most distant room from the agent. The agent receives an egocentric view of its surroundings, consisting of 3 × 3 pixels. The task increases in difficulty with X and Y. Episodes finish with a positive reward when the agent reaches the green goal square. Otherwise, episodes are terminated with zero reward after a maximum of 20 × N steps.

KeyCorridor. the agent has to pick up an object which is behind a locked door. The key is hidden in another room, and the agent has to explore the environment to find it. Episodes finish with a positive reward when the agent picks up the ball behind the locked door or after a maximum of 270 steps.

ObstructedMaze. The agent has to pick up a box which is placed in a corner of a maze. The doors are locked, the keys are hidden in boxes and the doors are obstructed by balls. Episodes finish with a positive reward when the agent picks up the ball behind the door or after a maximum of 576 steps.

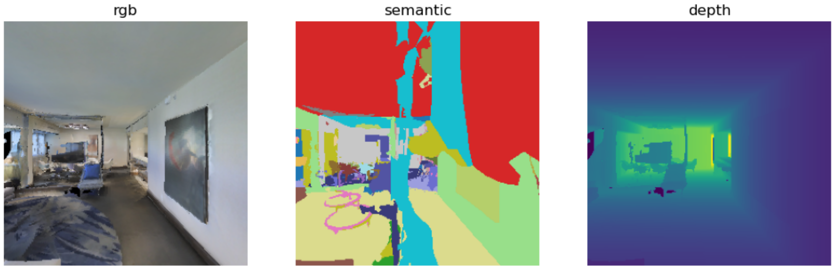


Figure 15: Visual observations in Habitat (RGB, semantic, and depth)

D.1.3 Habitat

The Habitat platform relies on several key abstractions that model the domain of embodied agents and the tasks they can perform in three-dimensional indoor simulation environments. This platform consists of a virtually embodied agent, such as a robot, equipped with a suite of sensors that can observe the environment and take actions to alter the agent's state or the environment's state. Each sensor is associated with a specific agent and can return observation data from the environment at a specified frequency. The 3D environment includes a scene mesh, objects, agents, and sensors, organized into regions and objects through a hierarchical representation called the Scene, which can be programmatically manipulated. All components of the Scene are present on

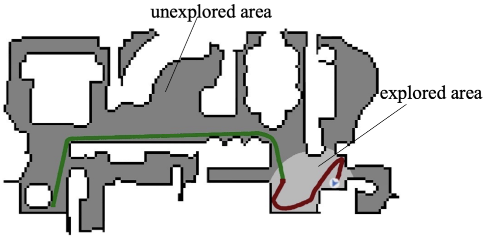


Figure 16: Explored area in Habitat

the SceneGraph. Additionally, a simulator backend instance can update the state of agents and SceneGraphs based on given actions for a set of configured agents and SceneGraphs, and it provides observations for all active sensors possessed by the agents. We use the Matterport3D (MP3D) dataset [15], which consists of high-quality renderings of indoor scenes. To measure exploration coverage, we compute the area revealed by the agent's line of sight using the function provided by the Habitat codebase [32]. As depicted in Figure 16, exploration is measured as the proportion of the environment revealed by the agent's line of sight over the course of the episode.

D.2 Algorithm Details

Algorithm 1 Effective Metric-based Exploration-bonus (EME)

```
Initialize parameters of policy $\pi(\theta)$, metric $d_E^{\phi}$, and reward models $\{g(\eta_1), \ldots, g(\eta_k)\}$
while not converged do
    for $t = 1$ to MAX_STEP_PER_EPISODE do
        Sample action $a_t \sim \pi_\theta\left(a_t \mid s_t\right)$
        Step environment $s_{t+1} \sim p\left(s_{t+1} \mid s_t, a_t\right)$
        Record transition in the buffer: $\mathcal{D}_\tau \leftarrow \mathcal{D}_\tau \cup \{s_t, a_t, s_{t+1}, r_{t+1}\}$
        Sample a mini batch with size $B \sim \mathcal{D}$ from the buffer
        Compute the variance $\zeta(r_{s_t})$ as scale factor from the ensemble of reward models
        Compute bonus $b_{t+1} = d_E(s_t, s_{t+1}) * min\{max\{\zeta(r_{s_t}), 1\}, M\}$
        Reshape reward with the bonus $r'_{t+1} = r_{t+1} + b$
        Train policy $\pi(\theta)$ via policy gradient
        Update the metric $d_E^{\phi}$: $\mathbb{E}_B[J(\phi)]$    ▷Equation (9)
        Update reward models by minimizing the prediction error: $\mathbb{E}_{\mathcal{D}_\tau} \|g(s_t, a_t; \eta) - r_{t+1}\|^2$
    end for
end while
```

D.3 Computation Details

All of our experiments are conducted on 4 GPUs with 16 CPU threads, which include AMD Ryzen 9 CPU@1.20GHz (16 core) CPU, NVIDIA GeForce GTX 3080Ti GPUs, and 64GB memory.

D.4 RL Hyperparameters

Table 6: Hyper-parameters in Robosuite experiments

Parameters		Environment	
	Pick and Place	Door Opening	Table Wiping
Horizon steps	300	300	500
Control frequency (Hz)	20	20	20
State dimension	110	22	30
Action dimension	4	8	7
Controller type	OSC position	Joint velocity	Joint velocity
Maximum reward scaling	10	10	20
Max training steps	1×10^7	-	-
Actor learning rate	3×10^{-4}	-	-
Critic learning rate	1×10^{-3}	-	-
Batch size	128	-	-
Discount factor	0.99	-	-
Soft update weight	0.005	-	-
Alpha learning rate	3×10^{-4}	-	-
Hidden layers	[256, 256, 256]	-	-
Returns estimation step	4	-	-
Buffer size	1×10^6	-	-
Steps per update	10	-	-

Table 7: Common Hyperparameters for MiniGrid

Parameter	Value
Learning Rate	0.0001
RMSProp momentum	0
RMSProp ϵ	10^{-3}
Unroll Length	100
Number of buffers	80
Number of learner threads	4
Number of actor threads	256
Max gradient norm	40
Entropy Cost	0.0005
Baseline Cost	0.5
Discounting Factor	0.99
Maximum reward scaling	5

Table 8: Common Hyperparameters for Habitat

Parameter	Value
Clipping	0.2
PPO epochs	2
Number of minibatches	16
Value loss coefficient	0.5
Entropy coefficient	5e-5
Learning rate	7e-5
ϵ	10^{-5}
Max gradient norm	0.8
Rollout steps	128
Use GAE	True
γ	0.99
τ	0.95
Use linear clip decay	True
Maximum reward scaling	30
Hidden size	512

Algorithm	Hyperparameters
Base Learner (PPO)	Maximum reward scaling: 10 threshold of probability ratio clipping (ϵ) : 0.5 update timesteps: 2e10 number of epoches per update: 50 number of minibatches: 16 batch size: 2048 GAE parameter (λ) : 0.95 optimizer: Adam learning rate: 5×10^{-3} policy gradient clip norm: 0.9 discount rate (γ) : 0.99
E3B	ridge regularizer λ: 0.1 tanh activation entropy cost: 0.005 intrinsic reward coefficient: 1.0 gradient clip norm: 1.0
ICM	forward model loss coefficient: 0.2 inverse model loss coefficient: 0.08 entropy cost: 10^{-4} intrinsic reward coefficient β: 0.005
RND	proportion of experience used for training predictor: 0.25 predictor Model updates per PPO epoch: 6 entropy cost: 0.005 intrinsic reward coefficient β: 0.1 forward model loss coefficient: 0.5
RIDE	inverse model loss coefficient: 0.8 entropy cost: 0.0005 intrinsic reward coefficient β: 0.1
NovelD	episodic memory capacity: 5000 action prediction network filter sizes: (3,3) entropy cost: 0.005 RND clipping factor L: 5 intrinsic reward coefficient β: 1.0
LIBERTY	intrinsic reward coefficient β: 0.05 entropy cost: 0.005 inverse model loss coefficient: 0.5 forward model loss coefficient: 0.5

Table 9: The hyperparameters of the tested benchmark algorithms in the Atari experiment

D.5 Codebases Used

Our codebase was built atop the following codebases:

- The official NovelD codebase: https://github.com/tianjunz/NovelD (Creative Commons Attribution-NonCommercial 4.0 license) for NovelID, RND, RIDE and count-based baselines (this codebase is built atop the official RIDE codebase below)
- The official RIDE codebase: https://github.com/facebookresearch/impact-driven-exploration (Creative Commons Attribution-NonCommercial 4.0 license)
- The official LIBERTY codebase: https://github.com/Mingle0228/liberty (MIT License)
- The official E3B codebase: https://github.com/facebookresearch/e3b(Creative Commons Attribution-NonCommercial 4.0 license)

E Limitation

While our method has shown success in several challenging exploration tasks, it does not consistently achieve top performance in environments with less sparse rewards, as observed in the MuJoCo continuous control benchmarks. Additionally, our method has not been tested in reinforcement learning domains characterized by large action spaces. Addressing these limitations and developing a more generalized solution for diverse RL environments remains a goal for future work.

F Related Work

Exploration in RL. Exploration remains a long-standing problem in RL. Common approaches include ϵ-greedy [59], count-based exploration [7, 46, 61, 41, 40, 57], ensemble-based exploration [45, 39, 48] and curiosity-based exploration [53, 55, 56, 10]. Several exploration strategies use a dynamics model to provide intrinsic rewards [47, 11, 33, 48, 36, 2]. Latent variable dynamics have also been studied for exploration [4, 9, 62]. Maximum entropy in the state representation has also been used for exploration [54, 66]. Other intrinsic motivation methods have recently been developed for exploration in context MDPs [67, 28, 51, 69, 32], which automatically generate curricula over variations of the MDP to encourage efficient learning, effectively performing a form of curiosity-driven exploration in the context space, including goal-conditioned [23, 50, 12, 18, 22] and goal-free variants [58, 49, 34, 19]. Reward shaping refers to modifying the original reward function with a shaping reward function which incorporates domain knowledge. Considering the most general form, namely the additive form, of reward shaping. The first approach to guarantee policy invariance is potential-based reward shaping (PBRS) [44], which defines the shaping reward function as the difference between values assessed through the potential function based on prior knowledge. There are lots of variants of PBRS [20, 38, 31, 3, 64, 30, 29, 63] which have shifted their focus to different areas within the field of reinforcement learning.

Metric-based Exploration Bonus. Metric space is widely used in encoding state representations [68, 14, 13, 16, 35]. Metric-based exploration bonus is based on the evaluation metric that quantifies the degree of difference or distance between two projected states for the measure of novelty. RIDE [51] evaluates the novelty as between two successive state representations under the L_2 norm. NovelD [69] uses the disparity in RND bonuses between adjacent states under the L_1 distance as the exploration bonus. LIBERTY [63] uses the difference between potential functions of adjacent states under the bisimulation metric space. Note that we propose an effective metric that measures the distribution distance between dynamics models by computing the distance between sampled subsequent states. Additionally, we integrate the Kullback–Leibler (KL) divergence between policy distributions to more robustly model the "**behavioral similarity**" between states, which is more effective and scalable across diverse environments with different observations, especially in addressing the critical Noisy-TV problem during exploration.

NeurIPS Paper Checklist

1. **Claims** Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?[Yes] In the introduction section (page 1).
2. **Limitations** Question: Does the paper discuss the limitations of the work performed by the authors?[Yes] In Appendix E.
3. **Theory Assumptions and Proofs** Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?[Yes] In the Appendix B.
4. **Experimental Result Reproducibility** Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?[Yes] In Appendix D.
5. **Open access to data and code** Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?[Yes] In the anonymous link https://anonymous.4open.science/r/EME-4F8B/README.md.
6. **Experimental Setting/Details** Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?[Yes] In Appendix D.4.
7. **Experiment Statistical Significance** Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?[Yes] In experiment section (page 5) and Appendix C.
8. **Experiments Compute Resources** Question: For each experiment, does the paper provide sufficient information on the computer resources (type of computing workers, memory, time of execution) needed to reproduce the experiments?[Yes] In Appendix D.3.
9. **Code Of Ethics** Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?[Yes]
10. **Broader Impacts** Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?[Yes] In Appendix A.
11. **Safeguards** Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?[NA]
12. **Licenses for existing assets** Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited, and are the license and terms of use explicitly mentioned and properly respected? [Yes] In Appendix D.5.
13. **New Assets** Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?[NA]
14. **Crowdsourcing and Research with Human Subjects** Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)? [NA]
15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects** Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?[NA]