Does behavioral diversity in intrinsic rewards help exploration?

Aya Kayal, Eduardo Pignatelli, Laura Toni
Department of Electronic and Electrical Engineering
University College London
London, UK

Abstract

In recent years, intrinsic reward approaches have attracted the attention of the research community due to their ability to address various challenges in Reinforcement Learning, among which, exploration and diversity. Nevertheless, the two areas of study have seldom met. Many intrinsic rewards have been proposed to address the hard exploration problem by reducing the uncertainty of states/environment. Other intrinsic rewards were proposed to favor the agent's behavioral diversity, providing benefits of robustness, fast adaptation, generalization and hierarchical learning. We aim to investigate whether pushing for behavioral diversity can also be a way to favor exploration in sparse reward environments. The goal of this paper is to reinterpret the intrinsic reward approaches proposed in the literature, providing a new taxonomy based on the diversity level they impose on the exploration behavior, and complement it with an empirical study. Specifically, we define two main categories of exploration: "Where to explore" and "How to explore". The former favors exploration by imposing diversity on the states or state transitions (State and State + Dynamics levels). The latter ("How to explore") rather pushes the agent to discover diverse policies that can elicit diverse behaviors (Policy and Skill levels). In the literature, it is unclear how the second category behaves compared to the first category. Thus, we conduct an initial study on MiniGrid environment to compare the impact of selected intrinsic rewards imposing different diversity levels on a variety of tasks.

1 Introduction

One of the main open problems in Reinforcement Learning is the exploration challenge [1]. When the environment rarely provides rewards as feedback, classical exploration strategies (e.g., epsilon-greedy, Thompson sampling, Boltzman distribution) fail to learn efficiently [2]. This is known as the hard exploration problem [3], which is challenging due to the sparsity of the reward [3]. Intrinsic rewards [4] have been proposed among the possible solutions to address this limitation [3, 5, 6]. They are a part of the larger notion of intrinsic motivation defined by [7] as the tendency to "seek out novelty and challenges, to extend and exercise one's capacity, to explore, and to learn". In RL, intrinsic rewards aim to provide the agent with a bonus, which either favors exploration by reducing the uncertainty of states [8, 9, 10, 11, 12, 13], or favors behavioral diversity, defined as learning meaningfully different trajectories/policies to solve the task [14, 15, 16, 17]. However, these two options are not mutually exclusive; behavioral diversity, which helps in robustness and fast adaptation to novel tasks, might be a way to improve exploration beyond its other benefits. In a comprehensive survey on exploration methods [3], intrinsic rewards were categorized between rewarding novel states and rewarding diverse behaviors. This study showed that there are methods [18, 14, 19, 20] actively encouraging behavioral diversity to improve exploration, suggesting a connection between the two areas. While extremely interesting, this categorization is overlooked in [3] because *i)* the survey covers the entire literature on

2nd Workshop on Agent Learning in Open-Endedness (ALOE) at NeurIPS 2023.

exploration, *ii)* authors believes that behavioral diversity is still a novel concept being developed. In this paper, we aim to go deeper in this analysis, understanding if mechanisms that favor diversity can also push for good exploration. In short, our goal is to answer the following question: "Do intrinsic rewards that push toward different levels of diversity lead to different exploration and different performance"?

To achieve this goal, in this paper we present a categorization of intrinsic rewards according to the diversity level they impose (state, state + dynamics, policy, skill), in addition to an initial empirical study to compare them on MiniGrid. Our empirical findings do not support the belief that behavioral diversity, often associated with enhanced robustness, aids exploration. The reason is that behavioral diversity focuses on distinguishing between different behaviors rather than visiting the states as uniformly as possible. Moreover, balancing between diversity and return is tricky as it depends on the structure of the environment and the task at hand. Too much diversity can hinder convergence and learning efficiency.

2 Related work

We now provide an overview of the literature focused on surveys of intrinsic rewards, first, and then cover the state-of-the-art on the empirical impact of different intrinsic rewards.

When looking at existing surveys [21, 22, 23, 3, 6, 5, 24], we are interested in understanding the categorizations existing so far. Intrinsic rewards were commonly classified between prediction error (curiosity), information gain, learning progress and state novelty methods. They all belong to "reward diverse states" category, also called "knowledge acquisition", because they aim to find new knowledge about the environment. Interestingly, some of the works [6, 23, 24, 3] have introduced a new category which consists of self-supervised acquisition of diverse skills/goals. [24] called this category "competence-based intrinsic motivation" and focused on goal-conditioned RL, with different types of goal representations and goal sampling strategies. [3] characterized this category as "reward diverse behaviors" and summarized methods learning diverse policies, along with evolution strategies. However, none of the surveys divided the classes that target exploration imposing different diversity levels, and none provided any empirical understanding of the exploration behavior. Motivated by this, we decide to study intrinsic rewards from a diversity aspect, and propose a different way to categorize them according to the diversity level they impose on the exploration behavior (state/dynamics/policy/skill).

We are now interested in the works provided in the literature aimed at benchmarking different intrinsic rewards. Few works have compared intrinsic rewards which belong to the category that favors visiting diverse states: [25] compared State Count, Random Network Distillation (RND) [12], Intrinsic Curiosity Module (ICM) [10], Reward Impact Driven Exploration (RIDE) [26] on MiniGrid environment. The study aimed to evaluate different design choices such as the impact of weighting and scaling intrinsic rewards on their performance, as well as the effect of using different neural network architectures. It was shown that reducing the number of parameters of neural network architectures deteriorates performance, and there is no clear winner between the scaling strategies. Another study [27] evaluated Pseudo-counts [8], RND, ICM and Noisy Networks [28] within the Arcade Learning Environment (ALE), and learned that none of these methods outperform the epsilon-greedy exploration. Authors advocated for better practices on empirical evaluation for exploration.

Other works have looked at different comparisons of intrinsic rewards: global vs episodic bonuses. Global bonuses are calculated using the entire training experience while episodic bonuses are calculated using the experience from the current episode. [29] found that episodic bonuses are more crucial than global bonuses to improve exploration in procedurally generated environments such as MiniGrid. A later study [30] found that episodic bonuses tend to yield better results in situations where there is minimal shared structure across various contexts in MiniHack, while global bonuses tend to be effective in cases where there is a greater degree of shared structure.

To the best of our knowledge, none of the empirical studies have compared the category that pushes towards diverse behaviors which we call "How to explore" to the category which pushes towards diverse states "Where to explore". Thus, we complement our survey with an initial empirical study aimed at understanding the impact of different levels of diversity (state/policy/behavior) on exploration in several MiniGrid environments.

3 Diversity levels imposed by intrinsic rewards

Inspired by the previous study [3], we divide intrinsic rewards into two categories: "Where to explore?" (Section 3.1) and "How to explore?" (Section 3.2) and analyze carefully each of these classes, as described in the following and summarized in Figure 1.

Figure 1: Categorization of the different levels of diversity incurred by intrinsic rewards for exploration in RL.

3.1 *"Where to explore?"*

This category, also called "reward novel states" in [3], encourages the agent to visit diverse states not previously encountered, pushing the agent to explore states where its knowledge is most limited. Moreover, we observe that the agent is pushed to acquire knowledge either from the states (State level diversity) or from both the state and the dynamics of the environment (State + Dynamics level diversity). In the following, we discuss in more details the works in each of these sub-categories:

State level diversity
In this subcategory, we collect all the works in which the intrinsic reward is a function of the state only, i.e., $r_{int} = f(s)$. The most common method is "State Count", which stores the visitation count of each state, and gives high intrinsic rewards to encourage revisiting states with low counts [31, 32, 33]. While counting works well in tabular cases, it becomes difficult in vast state spaces.

Several methods were proposed to extend State Count to large or continuous state spaces, such as pseudo-counts [8] and hashing [9].

Besides count-based methods, features prediction error can be used as a measure of the state novelty. For example, in [12], authors assessed state novelty by distilling a fixed randomly initialized neural network (target network) into another neural network (predictor network) trained on the data collected by the agent. This technique is called Random Network Distillation (RND), and the main motivation behind it is that the prediction error should be small for frequently visited states. Similarly, the NovelD algorithm [34] uses RND as a measure of state novelty but it defines the intrinsic reward as the difference in RND prediction errors at two consecutive states s_t and s_{t+1} in a trajectory.

Finally, this level of diversity includes methods which aim to maximize the entropy of the state distribution induced by the policy over finite or infinite horizon by estimating the state density distribution [11, 35] or by relying on the K-Nearest Neighbours (KNN) distance as approximation of state entropy [36, 37, 38].

State + Dynamics level diversity
This class also aims to visit diverse states but the difference with respect to State level is that the agent considers the novelty of the dynamics as well (not only states) to drive exploration. The agent either tries to build an accurate dynamical model of the environment or learns a dynamics-relevant state representation for exploration.

This subcategory mainly includes curiosity-driven methods which use the forward dynamics prediction error as intrinsic reward, such as [10] and [39]. The key intuition is to encourage the agent to revisit the unfamiliar state transitions where the prediction error is high (high mismatch between the expectation and true experience of the agent). Another curiosity-driven technique is Variational Information Maximizing Exploration (VIME) [40], which pushes the agent to explore states leading to a larger change in the dynamics model (higher information gain).

Moreover, this subcategory includes techniques that estimate the state novelty in a feature space which captures the temporal or dynamical aspect of states. For example, Exploration via Elliptical Episodic Bonuses (E3B) [41] and RIDE [26] both use an inverse dynamics model (ICM) to learn a state embedding that captures the controllable dynamics of the environment. While RIDE encourages the agent to take actions that significantly change the state embedding, E3B employs an elliptical episodic bonus. Other examples are Never Give up (NGU) [13], Agent 57 [42], and Episodic Curiosity (EC) [43]) which are all memory-based methods using a distance-based metric in a dynamical-aware feature space to approximate the state + dynamics novelty. Finally, authors in [44] use the inverse of the norm of the successor representation as intrinsic reward, capturing the transition dynamics.

3.2 *"How to explore?"*

We now explore the second category of intrinsic reward methods ("reward diverse behaviors" in [3]), which focuses on how to cover the state space, by favoring the visitation of states via diverse behaviors. This means that the agent is not driven by increasing the knowledge about states and environment directly (as in Section 3.1) but rather it is pushed by maximizing the diversity of the experienced behaviors. Also in this case, we identified two sub-categories depending on the level of diversity. Policy level diversity searches in the space of actions and aims to try different actions from given states. On the other hand, Skill level searches in the space of skills which are policies associated with latent variables (goal embeddings z / options Ω). It aims to acquire a repertoire of skills, which partition the state space into goals and learns policies to reach these goals.

Policy/Action level diversity
Algorithms in this subcategory aim to explore diverse actions from the same state. The intrinsic reward is a function of the policy here: $r_{int} = f(\pi(\cdot|s_t))$. What makes it different from the State + Dynamics algorithms introduced in Section 3.1 is that the previous category uses knowledge about states and dynamics of the environment, and pushes for exploring the areas where the agent knows the least (high uncertainty). In contrast, this level of diversity considers the previous exploration behavior represented by the policy (how the agent has explored) and pushes it to explore differently, inducing diversity on the policy learned. For example, in Maximum Entropy RL (Max Ent), the aim is to learn the optimal behavior while acting as randomly as possible. The objective function becomes the sum of expected rewards and conditional action entropy [16]. Soft-Actor Critic (SAC) [45] is a popular RL algorithm implementing the Max Ent RL framework. Diversity-driven exploration

strategy [18] is another exploration technique that encourages the agent to behave differently in similar states. It maximizes the divergence between the current policy and prior polices. Similarly, Adversarially Guided Actor-Critic (AGAC) [46] maximizes the divergence between the prediction of the policy and an adversary policy trained to mimic the behavioral policy. The main motivation is to encourage the policy to explore different behaviours by remaining different from the adversary. Another branch which belongs to this diversity level is the population-based exploration, which combines evolutionary strategies with Reinforcement Learning. These approaches train a population of agents to learn diverse behaviours which are high scoring at the same time, in order to effectively explore the environment [47, 48].

Skill level diversity
Skill level diversity disentangles diverse behaviors into different latent-conditioned policies (also called skills). The policy π is conditioned on a latent variable $z \sim p(z)$, and each z defines a different policy denoted by $\pi(a|s, z)$ [19]. This category aims to discover diverse skills and the intrinsic reward is a function of the skill: $r_{int} = f(z)$. Most methods falling in this category come from the domain of unsupervised skill discovery and use a discriminator-based architecture such as Diversity is all you need (DIAYN) [14]. DIAYN replaces the task reward with a learned discriminator term $q_\phi(z|s)$ that infers the behavior from the current state, in order to generate diverse policies visiting different set of states. It also uses Max Ent RL framework to learn skills which are as random as possible [14]. Maximum entropy diverse exploration (MEDE) [19] is very similar to "DIAYN + extrinsic reward", with the small difference of conditioning the discriminator on the state-action pair $q_\phi(z|s, a)$ instead of the state only. Moreover, MEDE uses the discriminator term as prior in the objective function instead of adding it as intrinsic reward. Structured Max Ent RL (SMERL) is another algorithm with the same approach as DIAYN, but it adds the intrinsic reward to the task reward, only when the policies have achieved at least near-optimal return [49]. DOMINO also learns diverse policies while remaining near optimal; it uses an intrinsic reward that maximizes the diversity of policies by measuring the distance between the expected features of the policies' state-action occupancies [15]. Finally, it's important to mention that skills in the literature can be called options or goals. Variational intrinsic control (VIC) is a framework which provides the agent with an intrinsic reward which relies on modelling options and learning policies conditioned on these options [17]. Instead of sampling options from a fix prior distribution as in DIAYN, VIC learns the prior distribution of options and updates it in order to choose options with higher rewards [17]. DIAYN and VIC are part of goal-conditioned RL methods, where goals are internally generated by agents and achieved via self-generated rewards [24].

4 Methodology of the empirical study

After describing in the previous section the different levels of diversity that can be imposed by intrinsic rewards, we proceed by describing the protocol of our experimental study which aims to gain an understanding of the differences between these levels of diversity on the exploration behavior.

4.1 Exploration algorithms

We augment the task reward with an intrinsic reward such that the total reward becomes: $r_{total} = r_{ext} + \beta * r_{int}$ where r_{ext} is the extrinsic reward defined according to the task, r_{int} is the intrinsic bonus and β is the hyper-parameter controlling the exploration-exploitation trade-off [3]. The best values of β can be found in Table 3 of Appendix C. We select four different intrinsic reward formulations representative of each diversity level:

1. State Count (State level diversity):
   It simply adds an intrinsic reward which is inversely proportional to the state visitation count [32]. For a transition (s_t, a_t, s_{t+1}), $r_{int} = 1/\sqrt{N(s_{t+1})}$ where $N(s_{t+1})$ is the number of times state s_{t+1} has been visited so far during training.
2. Intrinsic Curiosity Module ICM (State + Dynamics level diversity):
   This method uses curiosity as intrinsic reward to promote exploration. Curiosity is formulated as the error in the agent's ability to predict the outcome of its own actions in a learned state embedding space [10]. ICM trains a state embedding network, a forward and inverse dynamic models. For a transition tuple (s_t, a_t, s_{t+1}), the current state s_t and next state s_{t+1} are embedded into the features $\phi(s_t)$ and $\phi(s_{t+1})$ respectively, where ϕ is the embedding

network. Then, the inverse dynamics model $g : (\phi(s_t), \phi(s_{t+1})) \rightarrow \hat{a}_t$ takes as input the current and next state embeddings and predicts the action. The state embedding network is updated at the same time, such that the feature space only captures the aspects of the environment that are controlled by the agent's actions and ignores the uncontrollable factors. The forward dynamics model $f : (\phi(s_t), a_t) \rightarrow \hat{\phi}(s_{t+1})$ predicts the next state embedding given the current state embedding and current action. The intrinsic reward is generated using the prediction error of the forward dynamics model: $r_{int} = \|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2$ [10].

3. Max Ent RL (Policy level diversity):
The Maximum Entropy Reinforcement Learning framework [45] favors stochastic policies by augmenting the extrinsic reward with a policy entropy bonus $r_{int} = H(\pi(.|s_t))$ [50]. Hence, the agent seeks to maximize the following objective function: $J(\pi) = \mathbb{E}_{a_t \sim \pi}\left[\sum_{t=0}^{T} r(s_t, a_t) + \beta * H(\pi(.|s_t)\right]$, where T is the horizon, $r(s_t, a_t)$ is the extrinsic reward and β is the hyperparameter that calibrates the entropy term [45] .
4. DIAYN + Extrinsic Reward (Skill level diversity): Diversity is all you need (DIAYN) [14] learns a skill which is defined as a policy $\pi(a|s, z)$ conditioned on the state s and discrete latent variable z sampled from a prior distribution $p(z)$. This latent-conditioned policy captures diverse policies defined for each z. The goal is to partition the states between these skills and ensure that the skills are not only distinguishable from each others but as diverse as possible (maximum entropy). Thus, a discriminator $q_\phi(z|s)$ is trained to estimate the skill z from the state s, and an intrinsic reward $r_{int} = log(q_\phi(z|s)) - log(p(z))$ is used. An entropy regularizer takes care of maximizing the skills' entropy. Note that DIAYN is originally unsupervised, however, to ensure a fair comparison with other algorithms based on intrinsic-reward, we extend DIYAN to task-extrinsic reward in this study. At the start of each episode, a skill z is sampled from a fix uniform distribution $p(z)$, then the agent acts according to $\pi(a|s, z)$ and gets rewarded for collecting extrinsic rewards and visiting states that are easily distinguishable. Then the discriminator is updated to maximize the discriminability of skills, and the policy is updated to maximize the total reward (extrinsic and intrinsic) using any RL algorithm.

4.2 Environment

We test on MiniGrid [51], a widely used procedurally generated environment in RL exploration benchmarks [26, 25, 29]. Since we are interested in studying the impact of different diversity levels on exploration, we pick three commonly used sparse reward environments of MiniGrid, suitable to study behavioral diversity due to large grid sizes, open spaces, and strategical tasks.

1. Door Key 16×16: The agent has to pick up the key, then unlock the door to reach the goal.
2. Red Blue Doors: The agent has to open the red door then the blue door facing it on the opposite side.
3. Four Rooms: The agent has to navigate a maze of four rooms separated by gaps, to reach the goal randomly placed in one of the rooms.

In all three environments, the reward is collected at the end when the agent solves the task. The observations are partially observable and consist of a grid encoding of size $7 \times 7 \times 3$. More details about the tasks, observation, and action spaces are included in Appendix A.

4.3 Experimental Protocol

We test the four exploration algorithms introduced in Section 4.1 in a systematic comparative study. As the base learning algorithm to train the policy and value function [52], we choose the widely used Proximal Policy Optimization (PPO) with default hyperparameters [25] listed in Table 2 of Appendix C. This baseline algorithm comes with an entropy regularization in the objective function to encourage a minimum level of exploration [53], which is essential to avoid overfitting [54], and stabilize the training process [50]. The entropy regularization coefficient is set to $\epsilon = 0.0005$ in all simulated algorithms. We picked this value as large enough to guarantee a minimum level of convergence, but small enough to avoid to jeopardize the study on the impact of the intrinsic reward itself on exploration. The remaining four algorithms described in Section 4.1 are built by adding to

this baseline algorithm the four intrinsic rewards. The network architecture for the Actor-Critic model (PPO), as well as the state embedding network of ICM, and the discriminator network of DIAYN are all detailed in Appendix B. We train for a total number of 4×10^7 frames on all environments and we plot the training curves averaged over five runs with different seeds.

5 Experimental Results and Discussion

To evaluate the different intrinsic rewards, we plot the episodic return during training. We analyze which method reaches the maximum return the fastest. We also plot the observation coverage (number of visited partial observations), state coverage (number of visited (x, y) grid positions) and the entropy of the policy. We record the time steps at which the reward is found for the first, second and third time. Finally, we include plots of the mean intrinsic rewards during training and further visualizations (heatmaps) of the state visitation count ((x, y) positions) (see Appendices D and E for more detailed results). All plots correspond to training on procedurally generated MiniGrid environments for 40M frames, except for the state visitation heatmaps, we train on singleton environments for 10M frames to visualize the initial exploration behavior with a fixed map of the environments.

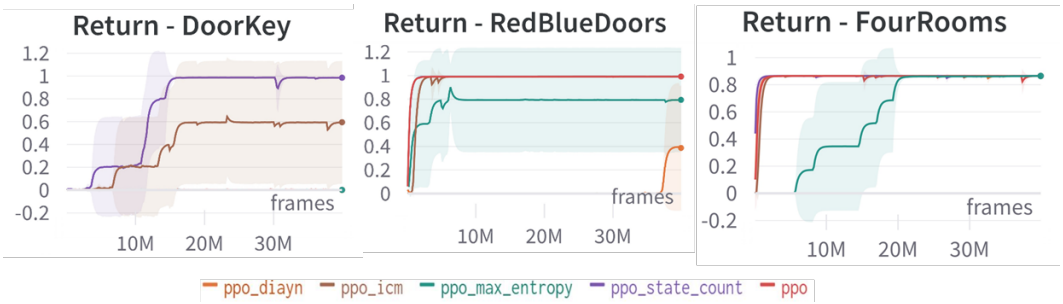


Figure 2: Mean episodic return during training for all three environments. The shaded region show the standard deviation between the different runs.

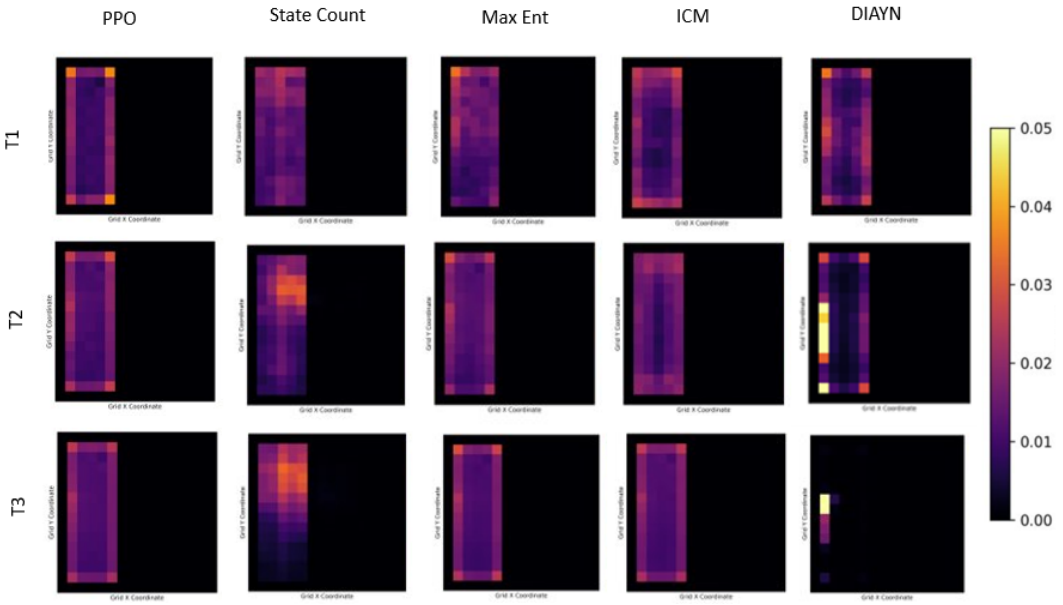


Figure 3: Normalised state visitation count during training for 10M frames on singleton DoorKey. For each intrinsic reward, snapshots of the heatmap are taken at three different timesteps T1: 100K frames, T2: 500K frames and T3: 10M frames. Refer to Figure 6a in Appendix A for the key and door positions.

In terms of extrinsic return (Figure 2), State Count performed the best on all three environments. On DoorKey, which is the hardest task between the three, State Count and ICM managed to converge, while the baseline PPO, Max Ent and DIAYN failed to solve it. On RedBlueDoors (easier environment to solve), the baseline PPO, State Count and ICM have all managed to perfectly solve it for all runs. Max Ent solved it, but had a worse average episodic return because it failed for one of the runs. DIAYN had the worst average return. Similarly, on FourRooms, all algorithms managed to solve the task, but Max Ent had a slightly worse average performance. The goal of our analysis is to understand the differences in the exploration behavior between the different intrinsic rewards and why do some methods explore better than others.

The high performance of State Count is explained by the fast coverage of the partial observations, as well as the (x, y) grid positions (see Figures 3 and 4 for the results on DoorKey environment). While PPO and Max Ent also covered the state space well on DoorKey, they did not cover it in a homogeneous way and did not manage to learn the connection between the key and the door, which prevented them from solving the task. State Count initially covered the state space more uniformly, which has led to a decay of intrinsic rewards (see Appendix D, Figure 8). This is confirmed by Figure 3 displaying the (x,y) positions in the grid occupied by the agent during training. This uniform exploration of State Count has also helped to converge faster, as shown from the entropy of the policy, which dropped quickly when the policy converges to a deterministic one (Figure 4). State Count was also the first intrinsic reward method to find the sparse extrinsic reward for the first, second and third time on DoorKey (Table 1). Regarding ICM, it showed a good performance and good state coverage but was not as consistent as State Count in solving the task. It had generally a lower convergence speed and higher standard deviation (shaded regions of ICM return, and policy entropy in Figures 2 and 4). This can be explained by the additional computational complexity of training the forward and inverse dynamics models. DIAYN and Max Ent did not manage to complete the task. The main reason of failure identifies so far are *i)* non-homogeneous exploration; *ii)* non-decreasing intrinsic reward, most likely as a consequence of the previous cause. The non-homogenous exploration is what to expect from the algorithm, that needs to discriminate diverse skills, but has a catastrophic effect, among which an intrinsic reward overcoming the extrinsic one and harming the overall performance. Max Ent also has a constant intrinsic reward and constant policy entropy; it was trying different actions (even unused ones in the task) and getting stuck in the corners where there are walls. Even if it found the reward for the first few times and covered most of the grid states, it did not learn to revisit the rewarding region due to the high non decreasing stochasticity. The phenomena of 'reward inflation' might have occurred here [55].

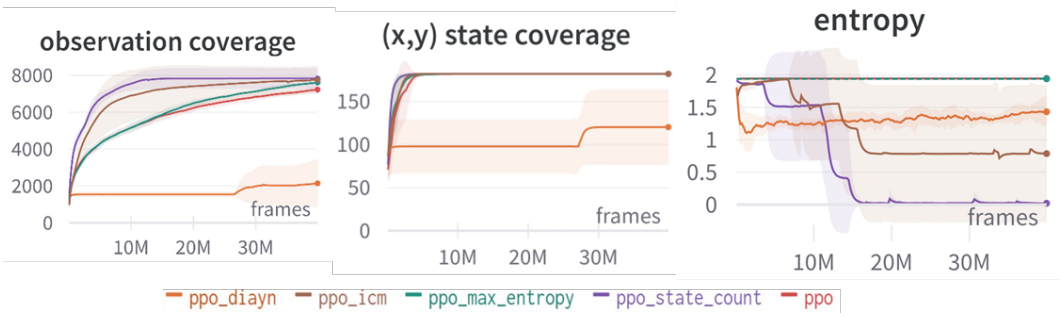


Figure 4: Mean observation coverage (grid encodings), state coverage (grid positions) and policy entropy during training on DoorKey. The shaded region shows the standard deviation.

The observed behaviors are confirmed on the other two environments (RedBlueDoors and FourRooms): Homogeneous exploration of State Count, similar exploration behavior but slower convergence of ICM due to predictor network training, unstable and non uniform exploration of Max Ent and DIAYN (see Appendices D and E). On RedBlueDoors (Section D.2 of Appendix D) , all intrinsic reward methods have covered the state and observation space well (no significant differences between them). Although Max Ent found the reward the soonest in this environment (even before State Count), the high value of the policy entropy destabilized the learning process and prevented the agent from solving the task in one of the runs. DIAYN has also focused on the states on the edge, then surprisingly, got stuck in the middle of the grid oscillating between four states (Figure 5). The

reason might be, as it was mentioned by [23], that skill learning methods (such as DIAYN) don't learn skills that cover the whole state space, and the discriminator can be restricted to a small area.

Table 1: Frame number at which the reward is found for the first, second, and third time by each exploration method on DoorKey environment. Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported.

DoorKey	First reward	Second Reward	Third reward
PPO	1114489.6 ± 542609.623	2014528 ± 939959.339	2626195.2 ± 1133921.092)
PPO + State Count	**496486.4 ± 550012.78)**	**558204.8 ± 548684.406**	**783075.2 ± 615917.672**
PPO + Max Ent	594649.6 ± 696956.504	1067401.6 ± 743704.05	3300668.8 ± 3108002.169
PPO + ICM	1089286.4 ± 734419.413	1287632 ± 674758.334	1683612.8 ± 539173.527)
PPO + DIAYN	>40M ± 12293415.172	>40M ± 12322129.862	>40M ± 12375101.418

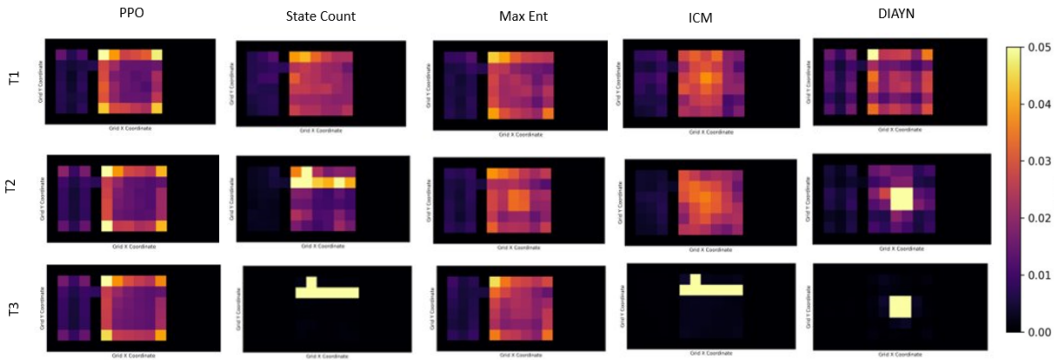


Figure 5: Normalised state visitation count during training for 10M frames on singleton RedBlueDoors. For each intrinsic reward, snapshots of the heatmap are taken at three different timesteps T1: 100K frames, T2: 500K frames and T3: 10M frames. Refer to Figure 6b in Appendix A for the map of the environment.

For FourRooms (Section D.3 of Appendix D), DIAYN exhibited the highest observation and state coverage. It's probably because the skills learned were not very diverse (discriminator loss not decreasing) and the environment is easy enough, so DIAYN was similar to the baseline PPO performance and solved the task. Max Ent seems to explore unnecessarily in this environment, which delayed finding the reward and resulted in a lower state coverage due to trying out all possible actions and getting stuck in some states (see Appendix E).

6 Perspectives and Conclusion

To sum up, in this work, we have revisited intrinsic reward techniques from the literature and proposed to classify them between State, State + Dynamics, Policy and Skill levels of diversity. We conducted empirical studies on MiniGrid, to understand the differences between them. Our results are limited to partially observable, not high dimensional state space and procedurally generated framework where each episode has a different context.

The first take home message is that the homogeneous exploration imposed by diversity on the State level (represented by State Count) has led to the best sample efficiency on many MiniGrid tasks. It improves the convergence speed in strategical tasks, has high state coverage and leads to a fast decrease of policy entropy and intrinsic reward. This decreasing rate of the intrinsic reward aligned well with finding the optimal behavior which avoided the dominance of the intrinsic reward. The solving time is another important consideration; methods using training networks (like ICM, and DIAYN) tend to be less sample efficient than simple State Count because the networks need to learn how to recognise different states/skills.

The second take home message is that DIAYN combined with extrinsic reward does not help in exploration more than State Count in the MiniGrid framework but might be helpful on other environments/frameworks. It's important to note that DIAYN was originally proposed as a completely

unsupervised skill pre-training method. In our study, we combined it with the extrinsic reward in order to compare it fairly to other intrinsic reward methods. We think that the bad performance of DIAYN might be due to the imbalance between diversity and reward maximization. DIAYN + Extrinsic reward is extremely sensitive to the hyperparameter β. This tradeoff between discriminability and optimality is a problem of discriminator-based architectures. Thus, finding the perfect linear or non linear combination between diversity and reward is crucial. In this case, DIAYN + Extrinsic reward might be promising to exploit some structures in the environment, which State Count and other intrinsic rewards might fail to explore. Discovering which settings / environment structures where behavioral diversity helps in exploration is still an open question.

References

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

[2] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

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

[4] Nuttapong Chentanez, Andrew Barto, and Satinder Singh. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17, 2004.

[5] Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke van Hoof, and Doina Precup. A survey of exploration methods in reinforcement learning. *arXiv preprint arXiv:2109.00157*, 2021.

[6] Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning. *arXiv preprint arXiv:1908.06976*, 2019.

[7] Richard M Ryan and Edward L Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68, 2000.

[8] 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.

[9] 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.

[10] 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.

[11] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.

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

[13] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

[14] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[15] Tom Zahavy, Yannick Schroecker, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. Discovering policies with domino: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521*, 2022.

[16] Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. *arXiv preprint arXiv:2103.06257*, 2021.

[17] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

[18] Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[19] Andrew Cohen, Lei Yu, Xingye Qiao, and Xiangrong Tong. Maximum entropy diverse exploration: Disentangling maximum entropy reinforcement learning. *arXiv preprint arXiv:1911.00828*, 2019.

[20] Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

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

[22] Nazmul Siddique, Paresh Dhakan, Inaki Rano, and Kathryn Merrick. A review of the relationship between novelty, intrinsic motivation and reinforcement learning. *Paladyn, Journal of Behavioral Robotics*, 8(1):58–69, 2017.

[23] Arthur Aubret, Laetitia Matignon, and Salima Hassas. An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy*, 25(2):327, 2023.

[24] Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey. *Journal of Artificial Intelligence Research*, 74:1159–1199, 2022.

[25] Alain Andres, Esther Villar-Rodriguez, and Javier Del Ser. An evaluation study of intrinsic motivation techniques applied to reinforcement learning over hard exploration environments. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 201–220. Springer, 2022.

[26] Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. *arXiv preprint arXiv:2002.12292*, 2020.

[27] Adrien Ali Taiga, William Fedus, Marlos C Machado, Aaron Courville, and Marc G Bellemare. On bonus-based exploration methods in the arcade learning environment. *arXiv preprint arXiv:2109.11052*, 2021.

[28] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295*, 2017.

[29] Kaixin Wang, Kuangqi Zhou, Bingyi Kang, Jiashi Feng, and YAN Shuicheng. Revisiting intrinsic reward for exploration in procedurally generated environments. In *The Eleventh International Conference on Learning Representations*, 2022.

[30] Mikael Henaff, Minqi Jiang, and Roberta Raileanu. A study of global and episodic bonuses for exploration in contextual MDPs. *arXiv preprint arXiv:2306.03236*, 2023.

[31] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

[32] 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.

[33] J Zico Kolter and Andrew Y Ng. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th annual international conference on machine learning*, pages 513–520, 2009.

[34] 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.

[35] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[36] Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. A policy gradient method for task-agnostic exploration. In *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020.

[37] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021.

[38] 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.

[39] 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.

[40] 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.

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

[42] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*, pages 507–517. PMLR, 2020.

[43] 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.

[44] 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.

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

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

[47] Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems*, 31, 2018.

[48] Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:18050–18062, 2020.

[49] Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured MaxEnt RL. *Advances in Neural Information Processing Systems*, 33:8198–8210, 2020.

[50] Jingbin Liu, Xinyang Gu, and Shuai Liu. Policy optimization reinforcement learning with entropy regularization. *arXiv preprint arXiv:1912.01557*, 2019.

[51] 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.

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

[53] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

[54] Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization. *arXiv preprint arXiv:1910.09191*, 2019.

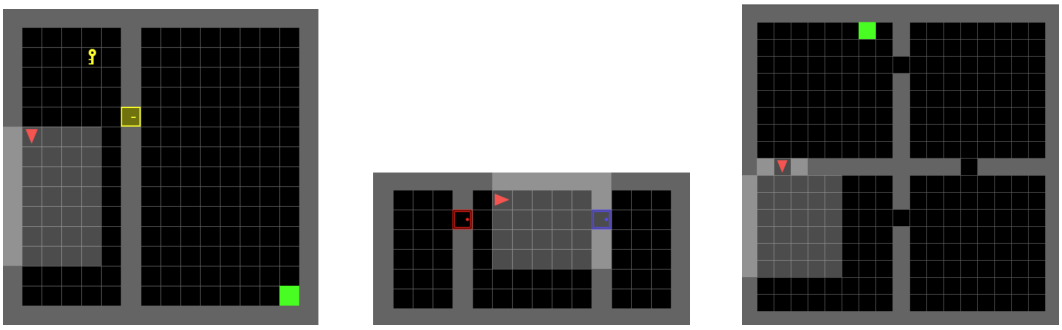
[55] Haonan Yu, Haichao Zhang, and Wei Xu. Do you need the entropy reward (in practice)? *arXiv preprint arXiv:2201.12434*, 2022.

[56] Jean-Baptiste Gaya, Laure Soulier, and Ludovic Denoyer. Learning a subspace of policies for online adaptation in reinforcement learning. *arXiv preprint arXiv:2110.05169*, 2021.

Appendices

A MiniGrid environments

1. DoorKey: This is a sparse reward environment which requires a certain order of visiting the states to solve the task; the agent needs to pick up the key, open the door then get to the green goal square. It does not get any reward after picking up the key or unlocking the door; it gets rewarded just at the end of the task. We use "MiniGrid-DoorKey-16x16-v0", which consists of a grid of size 16×16.
2. RedBlueDoors : The agent is randomly placed in a room where there are one red and one blue door facing opposite directions. The task consists of opening the red door before opening the blue door. The agent must rely on its memory of whether it has previously opened the other door to successfully complete the task, as it cannot see the door behind it. We use "MiniGrid-RedBlueDoors-8x8-v0".
3. FourRooms: In this environment, the agent must navigate a maze consisting of four rooms, with both its initial position and goal position being randomized. We use "MiniGrid-FourRooms-v0" where each of the four rooms consists of a grid of size 8×8.



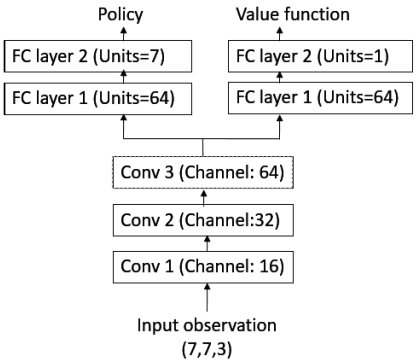
(a) Door Key 16×16 (b) Red Blue Doors (c) Four Rooms

Figure 6: MiniGrid environments

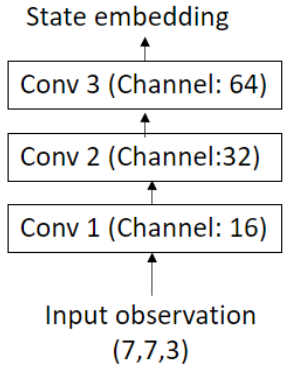
For all tasks, a maximum number of steps t_{max} is assigned, to encourage the agent to solve the task as fast as possible. When the agent succeeds after t steps, it gets a reward $r = 1 - 0.9t/t_{max}$ in all three environments. The episode ends when the agent collects the final reward or when the maximum number of steps is exceeded. By default, the observations are egocentric and partially observable. They consist of a grid encoding of size $7 \times 7 \times 3$. The first two dimensions (7×7) compose the tile set, and the last dimension encode the object type (wall, door, $\cdots$), the object color (red,green,$\cdots$) and the object status (door open, door closed, door locked). There are 7 actions available to the agent: turn left, turn right, move forward, pick up an object, drop an object, toggle and done. Some of these actions are unused in certain tasks.

B Neural Network Architectures

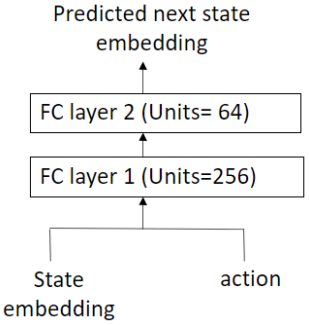
The network architecture for the Actor-Critic model is a shared 2D convolutional neural networks (CNN) to process the input observation followed by 2 separate heads (one head for the policy and one head for the value function). Each head is made of two fully connected layers. We use the same 2D CNN architecture to extract features in the state embedding network of ICM, and the discriminator network of DIAYN. All network architectures are represented in Figure 7.



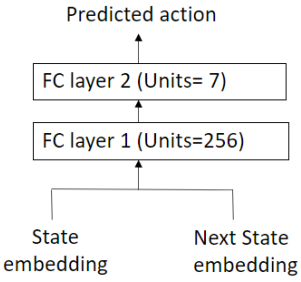
(a) Actor-Critic model



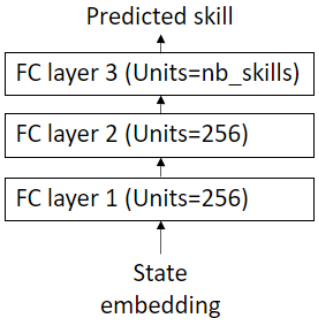
(b) State Embedding Network



(c) Forward dynamics model



(d) Inverse dynamics model



(e) Discriminator

Figure 7: Neural Network Architectures

C Hyperparameters

For State Count, and ICM, we use the hyperparameters of the previous study [25]. Since Max Ent + PPO and DIAYN + PPO were not tested before on MiniGrid, we run a grid search over $\beta \in [0.1, 0.01, 0.001, 0.0005]$ and pick the best values of β which result in the highest return during training. The chosen values of β are summarized in Table 3. For DIAYN, we choose to train 10 skills, which is the number used in the study by [56], and we use a discriminator learning rate of $3 \times e^{-4}$ following the implementation of DIAYN paper [14].

Table 2: List of hyperparameters

Number of parallel actors	16
Number of frames per rollout	128
Number of epochs	4
Batch size	256
Discount γ	0.99
Learning rate	0.0001
GAE λ	0.95
Entropy regularization coefficient	0.0005
Value loss coefficient	0.5
Clipping factor PPO	0.2
Gradient clipping	0.5
Forward dynamics loss coefficient	10
Inverse dynamics loss coefficient	0.1
Learning rates (state embedding, forward, and inverse dynamics)	0.0001
Number of skills	10
Discriminator learning rate	0.0003

Table 3: Best intrinsic reward coefficients β

	DoorKey	RedBlueDoors	FourRooms
State Count	0.005	0.005	0.005
Max Ent	0.0005	0.0005	0.0005
ICM	0.05	0.05	0.05
DIAYN	0.01	0.01	0.01

D Additional experimental results

D.1 DoorKey 16×16

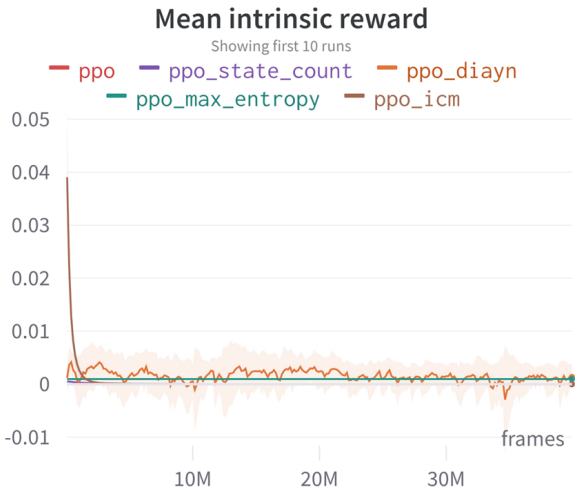
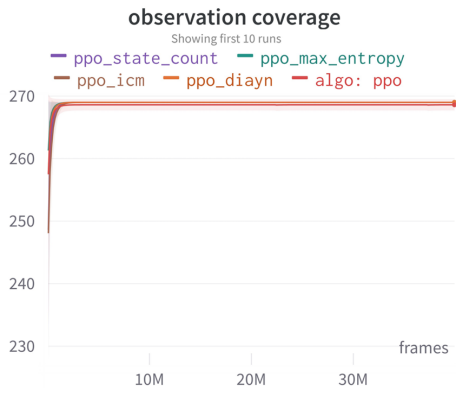


Figure 8: Mean intrinsic reward during training on DoorKey 16×16

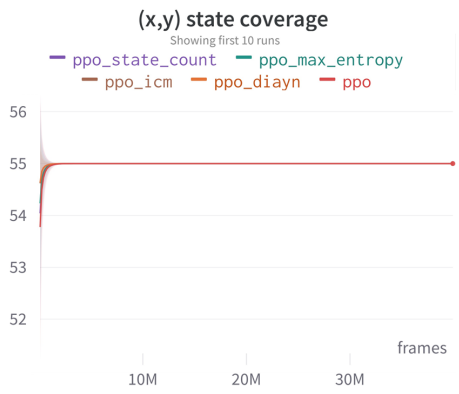
D.2 Red Blue Doors

Table 4: Frame number at which the reward is found for the first, second, and third time by each exploration method on Red Blue Doors environment. Results are averaged over five runs. Mean and standard deviation are reported.

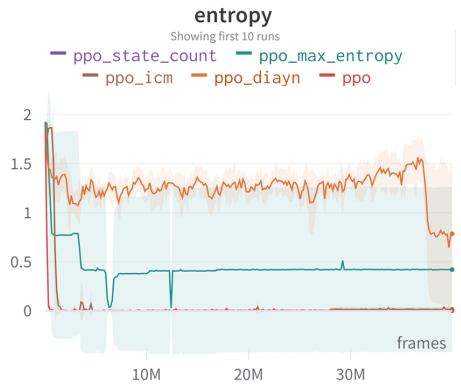
RedBlueDoors	First reward	Second Reward	Third reward
PPO	13136 $\pm$ 5647.717	**17568 $\pm$ 8303.114**	26553.6 $\pm$ 6733.478
PPO + State Count	13180.8 $\pm$ 8236.562)	25923.2 $\pm$ 14911.362	33545.6 $\pm$ 19115.737
PPO + Max Ent	**9417.6 $\pm$ 2678.856**	20464 $\pm$ 10420.749	**24432 $\pm$ 10339.3**
PPO + ICM	37721.6 $\pm$ 68636.525	129043.2 $\pm$ 175507.628	162060.8 $\pm$ 193005.435
PPO + DIAYN	10560 $\pm$ 11970.246	27712 $\pm$ 36493.862	44691.2 $\pm$ 37083.736



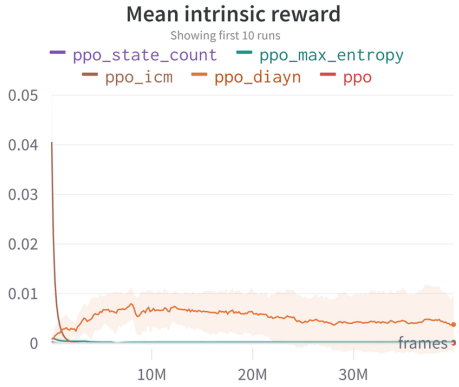
(a) Observation coverage for RedBlueDoors



(b) State coverage for RedBlueDoors



(c) Policy entropy for RedBlueDoors



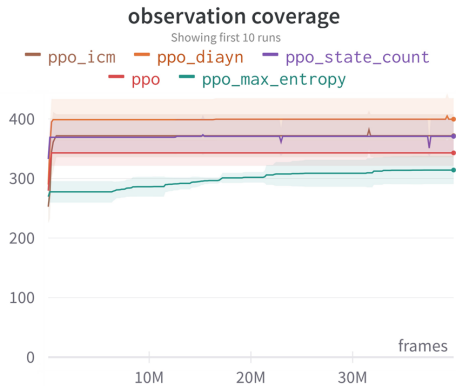
(d) Mean intrinsic reward for RedBlueDoors

Figure 9: Observation coverage, state coverage (grid position), policy entropy and mean intrinsic reward during training on RedBlueDoors

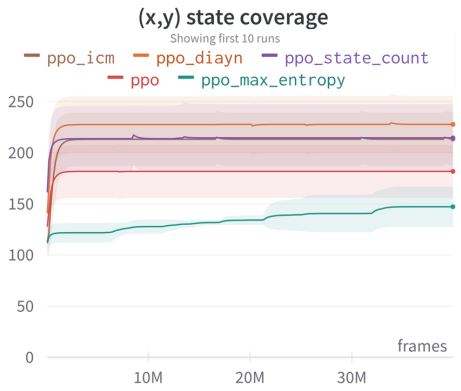
D.3 Four Rooms

Table 5: Frame number at which the reward is found for the first, second, and third time by each exploration method on Four Rooms environment. Results are averaged over five runs. Mean and standard deviation are reported.

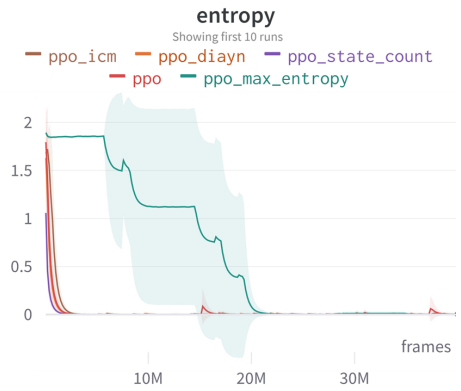
FourRooms	First reward	Second Reward	Third reward
PPO	29964 ± 35604.292	97033.6 ± 41446.258	150188.8 ± 104821.884)
PPO + State Count	**15465.6 ± 9712.017**	**34649.6 ± 11090.728**	**51820.8 ± 23054.047**
PPO + Max Ent	2479424 ± 5498212.722	5327913.6 ± 5306632.432	6874905.6 ± 5056693.48
PPO + ICM	89433.6 ± 111832.135	197312 ± 171435.209	274883.2 ± 171782.164
PPO + DIAYN	29872 ± 28094.241	91209.6 ± 111776.431	146348.8 ± 102613.828



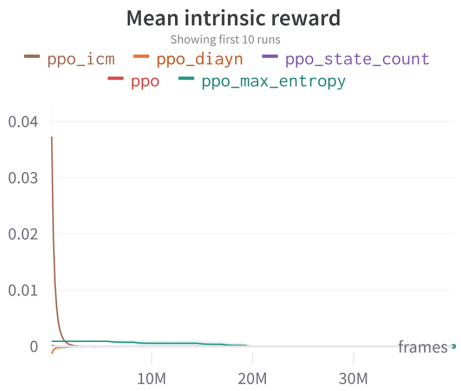
(a) Observation coverage for FourRooms



(b) State coverage for FourRooms



(c) Entropy for FourRooms



(d) Mean intrinsic reward for FourRooms

Figure 10: Observation coverage, state coverage (grid position), policy entropy and mean intrinsic reward during training on Four Rooms

E Additional heatmaps

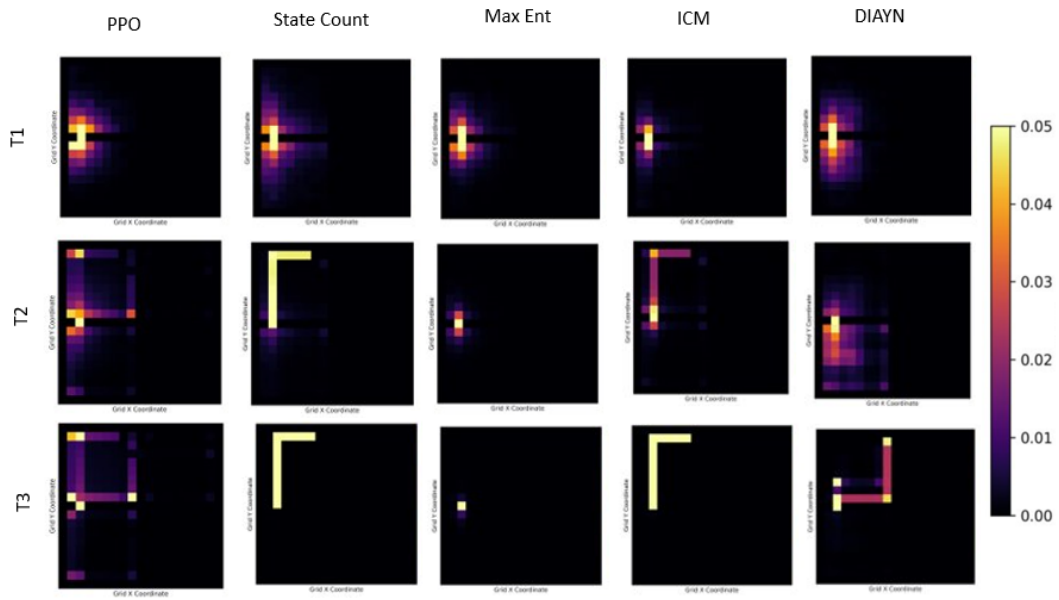


Figure 11: Normalised state visitation count during training for 10M frames on singleton FourRooms environment. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100K frames, T2: 500K frames and T3: 10M frames. Refer to Figure 6c in Appendix A for the map of the environment.