ORIGINAL ARTICLE

The impact of intrinsic rewards on exploration in Reinforcement Learning

Aya Kayal[1] · Eduardo Pignatelli[1] · Laura Toni[1]

Received: 11 August 2024 / Accepted: 4 May 2025 / Published online: 1 June 2025

Abstract
One of the open challenges in Reinforcement Learning (RL) is the hard exploration problem in sparse reward environments. Various types of intrinsic rewards have been proposed to address this challenge by pushing toward diversity. This diversity might be imposed at different levels, favoring the agent to explore different states, policies, or behaviors (State, Policy, and Skill level diversity, respectively). However, the impact of diversity on the agent's behavior remains unclear. In this work, we aim to fill this gap by studying the effect of different levels of diversity imposed by intrinsic rewards on the exploration patterns of RL agents. We select four intrinsic rewards (State Count, Intrinsic Curiosity Module (ICM), Maximum Entropy, and Diversity is All You Need (DIAYN)), each pushing for a different diversity level. We conduct an empirical study on MiniGrid environments to compare their impact on exploration considering various metrics related to the agent's exploration, namely: episodic return, observation coverage, agent's position coverage, policy entropy, and timeframes to reach the sparse reward. The main outcome of the study is that State Count leads to the best exploration performance in the case of low-dimensional observations. However, in the case of RGB observations, the performance of State Count is highly degraded mainly due to representation learning challenges. Conversely, Maximum Entropy is less impacted, resulting in a more robust exploration, despite not always being optimal. Lastly, our empirical study revealed that learning diverse skills with DIAYN, often linked to improved robustness and generalization, does not promote exploration in MiniGrid environments. This is because: (i) Learning the skill space itself can be challenging, and (ii) exploration within the skill space prioritizes differentiating between behaviors rather than achieving uniform state visitation.

Keywords Reinforcement learning · Intrinsic motivation · Exploration · Diversity

1 Introduction

The sparsity of rewards is a major hurdle for RL algorithms [1]. With infrequent feedback, the probability that the agent discovers a rewarding sequence of actions on a random basis becomes low. Therefore, a large number of samples are needed to explore and stumble into a successful sequence of actions leading to the desired outcome [2]. This is known as the hard exploration problem [1]. Classical exploration strategies, for example, epsilon-greedy and Boltzmann distribution [3] fail to explore the environment efficiently enough to find the optimal solution when the feedback is sparse [1, 4]. Among the possible solutions to address this limitation [1, 5], intrinsic rewards [6–8] have been proposed. They are a part of the larger notion of intrinsic motivation defined by Ryan and Deci [9] as the tendency to "*seek out novelty and challenges, to extend and exercise one's capacity, to explore, and to learn.*" Intrinsic rewards are often categorized in the literature into: *knowledge-based* and

competence-based [8, 10–12]. The first category encourages the agent to gain new knowledge about the environment. It compares the agent's experiences to its existing knowledge and rewards the agent for encountering unexpected situations. This includes methods that reward novelty in states or state transitions [13–16], the prediction error [17], or the information gain [18]. The second category, also called "*skill-learning*" in Aubret et al. [8, 19], rewards the agent for learning a diverse repertoire of skills in an unsupervised way. It mainly includes goal-conditioned RL approaches, which generate and achieve their own goals to explore the environment [20–22]. In Colas et al. [11], a detailed survey on goal-conditioned RL is presented, highlighting the different types of goal representations and goal-sampling strategies.

This categorization uncovers a potential link between diversity and exploration, where intrinsic rewards promote diverse agent behaviors to efficiently explore the environment. While diversity is acknowledged as crucial in RL, it has mainly been explored in relation to robustness, generalization, hierarchical learning, or generation tasks [23–30]. However, its role in driving effective exploration remains underexplored and not empirically validated yet. In this work, we take an initial step toward understanding whether mechanisms that encourage diversity through skill discovery can also drive more effective exploration. To address this gap, we propose a rigorous methodology to empirically compare *knowledge-based* and *competence-based* intrinsic rewards by further dividing them into distinct diversity levels (State, State + Dynamics, Policy, and Skills) and testing them in a unified empirical framework with various exploration metrics. Our work focuses on examining how different levels of diversity impact exploration, driven by the need to address the following open questions: (i) What is, in practice, an effective exploration in environments with low- and high-dimensional state spaces? (ii) How does the level of diversity imposed by intrinsic rewards affect exploration performance across different scenarios? (iii) Does behavioral diversity through skill discovery, known to help robustness and fast adaptation [25, 26], also help exploration?

2 Related works

While numerous intrinsic reward formulations have been proposed to address complex sparse reward tasks, a comprehensive understanding of their comparative advantages and challenges remains unclear. Here, we review previous works that have attempted to categorize or empirically compare intrinsic rewards. Table 1 provides an overview of these studies, highlighting the pros and cons of each approach. Existing surveys [1, 5, 8, 11, 12, 19, 31] offer slightly different taxonomies of intrinsic rewards, often using varied terminology. However, most include two broad categories: one focused on increasing knowledge about the environment (e.g., prediction error, information gain, learning progress, and state novelty), and another focused on learning diverse skills. Yet, these surveys lack empirical validation and none of them explore the different levels of diversity that these intrinsic rewards can introduce within each category.

We are now interested in the works provided in the literature aimed at benchmarking different intrinsic rewards. A few studies have compared methods within the *knowledge-based* category. For instance, Andres et al. [32] compared State Count [33], Random Network Distillation (RND) [15], Intrinsic Curiosity Module (ICM) [17], and Reward Impact Driven Exploration (RIDE) [34] on MiniGrid environment. The study aimed to evaluate the impact that weighting intrinsic rewards has on performance, as well as the effect of using different neural network architectures. Another study by Taiga et al. [35] evaluated pseudo-counts [13], RND, ICM, and Noisy Networks [36] within the Arcade Learning Environment (ALE) [37]. Their findings suggested that none of these methods outperform the epsilon-greedy exploration. A more recent work by Yuan et al. [38] introduced RLeXplore, a comprehensive plug-and-play framework that implements ICM [17], RND [15], Disagreement [39], Never Give Up (NGU) [16], PseudoCounts [13], RIDE [34], Random Encoders for Efficient Exploration (RE3) [40], and Exploration via Elliptical Episodic Bonuses (E3B) [41]. Their framework addressed critical design, implementation, and optimization issues related to intrinsic rewards. These included reward and observation normalization, co-learning dynamics of policies and representations, weight initialization, and the

Table 1 Comparison of existing works on intrinsic reward categorizations and empirical studies outlining the pros and cons

Study	Paper category	Pros	Cons/limitations
Ladosz et al. [1]	Categorization of intrinsic rewards	Categorized intrinsic rewards into reward novel states and reward diverse behaviors	Lack of empirical testing in a common framework
Aubret et al. [8]	Categorization of intrinsic rewards	Categorized intrinsic rewards into knowledge acquisition and skill-learning	Lack of empirical testing in a common framework
Aubret et al. [19]	Categorization of intrinsic rewards	Categorized intrinsic rewards into surprise, novelty and skill-learning	Lack of empirical testing in a common framework
Amin et al. [5]	Categorization of intrinsic rewards	Categorized intrinsic rewards into blind, uncertainty, space coverage and self-generated goals	Lack of empirical testing in a common framework
Colas et al. [11]	Categorization of intrinsic rewards	Categorized intrinsic rewards into *knowledge* and *competence-based* (focused on goal-conditioned RL)	Lack of empirical testing in a common framework
Hao et al. [31]	Categorization of intrinsic rewards	Categorized intrinsic rewards into prediction error, novelty, and information gain	Lack of empirical testing in a common framework
Siddique et al. [12]	Categorization of intrinsic rewards	Categorized intrinsic rewards into *knowledge-based* and *competence-based*	Lack of empirical testing in a common framework
Andres et al. [32]	Empirical study on intrinsic rewards	Evaluated performance of *knowledge-based* intrinsic rewards (State Count, RND, ICM, and RIDE) on MiniGrid. Analyzed different weighting methods for intrinsic rewards and different neural network architectures	Did not include any skill-learning methods from the *competence-based* category. Focused on return performance without directly studying exploration
Taiga et al. [35]	Empirical study on intrinsic rewards	Compared the performance of *knowledge-based* intrinsic rewards (Pseudo-counts, RND, ICM, and Noisy Networks) on ALE environment	Did not include any skill-learning/goal-conditioned methods. Evaluated performance solely based on return with no specific focus on the exploration behavior
Yuan et al. [38]	Empirical study on intrinsic rewards	Compared the performance of *knowledge-based* intrinsic rewards (ICM, RND, Disagreement, NGU, PseudoCounts, RIDE, RE3, and E3B). Addressed key design and optimization details of intrinsic rewards to establish standardized implementations	Did not include skill-learning/goal-conditioned methods. Did not provide any link between diversity and exploration
Laskin et al. [42]	Categorization and empirical study on intrinsic rewards	Tested methods from *knowledge-based* (ICM, Disagreement, and RND), *competence-based* (DIAYN, SMM, and APS) and *data-based* (APT and ProtoRL) categories on continuous control tasks. Evaluated their generalization capabilities in an unsupervised pretraining followed by supervised finetuning framework	Did not consider the joint optimization of intrinsic and extrinsic rewards. Focused on generalization and fast adaptation rather than studying the impact of diversity on exploration
Wang et al. [43]	Categorization and empirical study on intrinsic rewards	Divided intrinsic rewards between lifelong (global) and episodic bonuses. Tested different combinations of global and episodic bonuses on MiniGrid, in sparse reward and pure exploration settings. Analyzed why lifelong intrinsic reward does not contribute much in improving exploration	Focused on global versus episodic perspective, not on diversity and its impact on exploration
Henaff et al. [44]	Categorization and empirical study of intrinsic rewards	Studied the advantages and disadvantages of global and episodic intrinsic rewards for exploration in contextual MDPs	Interpreted intrinsic rewards from the global/episodic perspective, but not from a diversity perspective
Lin and Jabri [46]	General framework unifying intrinsic rewards	Interpreted intrinsic rewards as special cases of conditional prediction with different mask distributions	Lack of a comparative empirical study
Zahavy et al. [47]	General framework unifying intrinsic rewards	Reformulated the convex MDP problem as a convex-concave game and interpreted several RL algorithms (including skill-based intrinsic rewards) as instances of it	Lack of a comparative empirical study

combined optimization of intrinsic and extrinsic rewards. The study most similar to ours is by Laskin et al. [42], which evaluated intrinsic rewards across knowledge, competence, and data-based categories on the DeepMind Control Suite. However, their primary objective was to assess the generalization of unsupervised RL algorithms by measuring how quickly they adapted to diverse downstream tasks. To achieve this, they used a reward-free pretraining phase followed by supervised finetuning. In contrast, our study focuses on the standard RL setting, where both intrinsic and extrinsic rewards are optimized simultaneously (except for skill-based learning). Instead of concentrating on adaptation, we address the exploration challenge, evaluating intrinsic rewards from a diversity perspective and employing various metrics to measure exploration quality.

Other works have examined a different taxonomy of intrinsic rewards: global versus episodic bonuses. Global bonuses are calculated using the entire training experience, while episodic bonuses are calculated using only the experience from the current episode. The work by Wang et al. [43] found that episodic bonuses are more crucial than global bonuses to improve exploration in procedurally generated environments such as MiniGrid. A later study by Henaff et al. [44] found that episodic bonuses tend to yield better results in situations where there is minimal shared structure across various contexts in MiniHack [45], while global bonuses tend to be effective in cases where there is a greater degree of shared structure.

Additionally, some works aimed to unify different intrinsic reward formulations under a general framework. For instance, Lin and Jabri [46] proposed a unified framework for intrinsic rewards, showing that existing methods can be viewed as special cases of conditional prediction with different mask distributions. Building on this, they introduced a novel trajectory-level exploration intrinsic reward, which extends beyond the typical one-step future prediction to capture transition dynamics across longer time horizons. In a related line of work, Zahavy et al. [47] reformulated the convex MDP problem as a convex-concave game between an agent and an adversarial player generating costs (negative rewards). They unified a broad range of RL algorithms, including methods for unsupervised skill discovery, by interpreting them as instances of this generalized game-theoretic framework.

3 Novelty and scientific contributions

Despite significant progress in categorizing, evaluating, and interpreting intrinsic rewards in RL, a key gap remains: to understand how diversity in intrinsic rewards impacts exploration. Existing theoretical surveys often lack empirical validation, while empirical studies have yet to compare *competence-based* (skill-learning) and *knowledge-based* intrinsic rewards in terms of exploration behavior. Crucially, no prior work has systematically examined the role of diversity in exploration. We address this by evaluating intrinsic rewards from both categories across multiple MiniGrid environments.

Our key contributions are as follows:

1. We introduce a refined categorization of intrinsic rewards based on four diversity levels—State, State + Dynamics, Policy, and Skill—offering a more granular understanding of their influence on exploration.
2. We design an empirical study that assesses how these different diversity levels impact exploration by incorporating multiple complementary exploration metrics such as return, coverage, entropy, reward findings, and state visitation maps.
3. We provide empirical insights into the role of diversity in exploration, offering practical guidance to leverage intrinsic rewards for environments with varying exploration challenges.

Table 2 Best intrinsic reward coefficients β

	Empty	DoorKey	RedBlueDoors	FourRooms
State count	0.005	0.005	0.005	0.005
Max entropy	0.0005	0.0005	0.0005	0.0005
ICM	0.05	0.05	0.05	0.05
DIAYN	0.01	0.01	0.01	0.01

To achieve this, we evaluate representative intrinsic reward methods from each diversity level on MiniGrid [48], using both grid encodings and RGB observations. This setup allows us to analyze how diversity shapes agent behavior in exploration-critical environments. To the best of our knowledge, this is the first systematic evaluation of diversity levels in intrinsic rewards within a unified framework, offering novel insights into their influence on exploration and performance.

4 Methodology

In the following, we subclassify the *knowledge* and *competence-based* intrinsic reward methods according to the level of diversity they impose on the agent's exploration (Sect. 4.1). Then, we select four intrinsic rewards, one for each level (Sect. 4.2), and we test them empirically on MiniGrid environment, explained and motivated in Sect. 4.3. Section 4.4 outlines the experimental protocol used in the study, while Sect. 4.5 details the model architecture. Finally, Sect. 4.6 introduces the evaluation metrics.

4.1 Taxonomy of diversity levels imposed by intrinsic reward

We systematize the types of diversity imposed by intrinsic rewards into four levels: **State level diversity** encourages exploration of unseen states, pushing the agent toward areas where its knowledge is most limited. **State + Dynamics level diversity** also focuses on diverse states, but additionally considers the novelty of the dynamics between those states for a more comprehensive exploration. **Policy level diversity** explores the impact of different actions from given states, while **Skill level diversity** explores the effectiveness of diverse skills (policy-goal association) in achieving goals [11]. For a more detailed description of these diversity levels, please refer to Appendix A.

4.2 The selected intrinsic reward 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, r_{int} is the intrinsic reward, and β is the intrinsic reward coefficient [1]. The best-performing β values, either sourced from the literature [32] or determined through a grid search (details provided in Appendix C), are presented in Table 2, also located in Appendix C. We select four different intrinsic reward methods, each representative of one of the four diversity levels:

State Count (State level diversity) builds an intrinsic reward inversely proportional to the state visitation count [33]. For a transition (s_t, a_t, s_{t+1}), where s_t is the current state, a_t is the current action and s_{t+1} is the next state, $r_{int}(t) = 1/\sqrt{N(s_{t+1})}$, where $N(s_{t+1})$ represents the number of times s_{t+1} has been visited during training. This algorithm considers only discrete, low-dimensional state space. However, for RGB observations, where the state space is much larger and State Count is not feasible, we use SimHash [14] to hash states before counting them. SimHash maps the pixel observations to hash codes according to the following equation, with h as the hashing function: $h(s_{t+1}) = sgn(A * \phi(s_{t+1})) \in \{-1, 1\}^k$. Here, ϕ is an embedding function, A is a matrix with i.i.d. entries drawn from a standard normal distribution, k is the size of the hashed key, and $sgn(\cdot)$ maps a number to its sign. Then, the same intrinsic reward formula is applied but using the hashed observation: $r_{int}(t) = 1/\sqrt{N(h(s_{t+1}))}$.

Intrinsic Curiosity Module (ICM) (State + Dynamics level diversity) uses curiosity as an intrinsic reward. 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 [17]. Specifically, ICM trains a state embedding network, a forward and an inverse dynamic model. For a transition tuple (s_t, a_t, s_{t+1}), the embedding network $\phi : \mathcal{S} \rightarrow \mathcal{F}$ projects the current state s_t and next state s_{t+1} into the feature space $\mathcal{F}$ to get the embeddings $\phi(s_t)$ and $\phi(s_{t+1})$ respectively. Then, the

inverse dynamics model $g : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{A}$ takes as input the current and next state embeddings, $\phi(s_t)$ and $\phi(s_{t+1})$ respectively, and predicts the action a_t taken by the agent to move from state s_t to state s_{t+1}. The state embedding network is updated, such that it only captures the features of the environment that are controlled by the agent's actions, and ignores the uncontrollable factors. The forward dynamics model $f : \mathcal{F} \times \mathcal{A} \rightarrow \mathcal{F}$ predicts the next state embedding $\phi(s_{t+1})$ given the current state embedding $\phi(s_t)$ and current action a_t. The intrinsic reward is the prediction error of the forward dynamics model: $r_{int}(t) = \|f(\phi(s_t), a_t) - \phi(s_{t+1})\|_2^2$ [17].

Max Entropy RL (Policy level diversity) augments the extrinsic reward with the policy entropy $r_{int}(t) = H(\pi(.|s_t))$ to favor stochastic policies [49, 50].

DIAYN (Skill level diversity) aims to discover a set of diverse skills without supervision [20]. A skill is defined as a policy $\pi(a|s, z)$ conditioned on the state s and latent variable/goal z. DIAYN's objective is to maximize the mutual information between z and every state in the trajectory generated by $\pi(a|s, z)$. The intuition is to infer the skill from the state. At the start of each episode, a latent variable z is sampled from a uniform distribution $p(z)$, then the agent acts according to that skill $\pi(a|s, z)$ throughout the episode. A discriminator $q_\alpha(z|s)$ is trained to estimate the skill z from the state s. The intrinsic reward, defined by $r_{int}(t) = log(q_\alpha(z|s_{t+1})) - log(p(z))$, is used to push the agent to visit states that are easily distinguishable in terms of skills. Then, the discriminator is updated to better predict the skill, and the policy is updated to maximize r_{int} using any RL algorithm. It is worth mentioning that DIAYN has been proposed as an unsupervised skill discovery method to favor robustness, fast adaptation to new tasks and hierarchical learning. Therefore, exploration is not the main goal of DIAYN. As a consequence, DIAYN's intrinsic reward can conflict with the agent's extrinsic reward, potentially jeopardizing convergence if combined directly. To address this, we split the training budget between pretraining and finetuning phases. During pretraining, skills are learned using only intrinsic rewards. The learned weights are then used to initialize the policy and value networks for the finetuning phase with task-specific extrinsic rewards. The rationale for this approach is further explained in Appendix E, where we evaluate the performance of DIAYN when combined with extrinsic rewards.

4.3 Environment

We test on MiniGrid [48], a widely used procedurally generated environment in RL exploration benchmarks [32, 34, 43] suitable for experimenting with sparse rewards. We consider two types of observations: partially observable grid encodings, and partially observable RGB images (see Appendix B). The latter has a much larger

Table 3 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
SimHash key size K	16

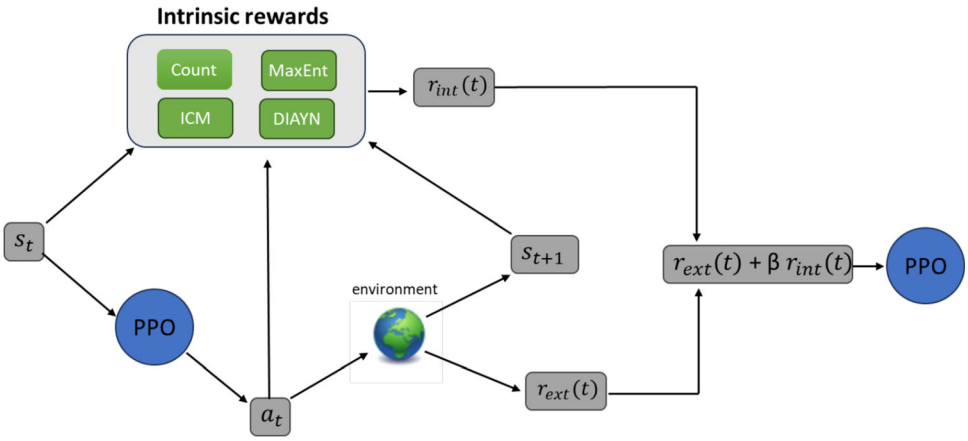


Fig. 1 Overview of the empirical study pipeline, illustrating the flow from input observations to action selection, and reward computation (both extrinsic and intrinsic) within the PPO framework

state space, allowing us to investigate the scenarios challenging for State level diversity algorithms. To study the impact of different diversity levels on exploration, we select four environments with varying grid layouts and tasks, which highlight the strengths and weaknesses of various intrinsic reward methods:

1. *Empty*: We choose this environment as a control and it is the only one not procedurally generated (fixed initial and goal positions). The setup imposes minimal constraints, providing freedom to solve the task in different ways. Consisting of one big homogeneous room, this environment is interesting since it can lead to state aliasing: Different MDP states, for example, different (x, y) positions of the agent, appear as identical observations [51]. This creates a challenge for state count-based methods, which count the observations they encounter and therefore cannot differentiate between the true underlying states.
2. *DoorKey*: This environment requires strategic exploration to locate keys and unlock doors. It stresses the importance of a trajectory to visit states in particular order. Methods that can learn skills and recognize these dependencies might perform better than state count-based methods, which treat all state visits equally without taking into account the order.
3. *FourRooms*: This environment is characterized by its sparsity of rewards. The presence of multiple rooms encourages the agent to devise different strategies for navigation, fostering diversity in the trajectories or paths taken by the agent to achieve the goal.
4. *RedBlueDoors*: This environment also requires strategic exploration, but it is an easier task than DoorKey. It introduces color-coded doors, requiring agents to exhibit high levels of cognitive flexibility.

More details about the tasks, observation, and action spaces are included in Appendix B.

4.4 Experimental protocol

We test the four algorithms in each environment for each observation space. We select proximal policy optimization (PPO) [52] as our baseline algorithm. PPO is a widely accepted and popular choice in RL research, known for its stability, robustness, and relatively high sample efficiency. Its simple implementation offers manageable computational costs, which enhances reproducibility and facilitates validation of results. Beyond its theoretical strengths, PPO has demonstrated success in complex real-world applications, such as large language model (LLM) research, underscoring its versatility and reliability for our study.

We adopt the default hyperparameters from [32], listed in Table 3 of Appendix C. This baseline algorithm comes with an entropy regularization in the objective function to encourage a minimum level of exploration [53]. Such regularization is essential to avoid overfitting [54] and to stabilize the training process [50]. We set the entropy regularization coefficient to $\epsilon = 0.0005$ in all simulated algorithms. The selected value is large enough to guarantee a minimum level of stable convergence but small enough to not affect our experiments. We train each algorithm for $40M$ frames on all environments. For DIAYN exclusively, we use $25M$ for pretraining and $15M$ for

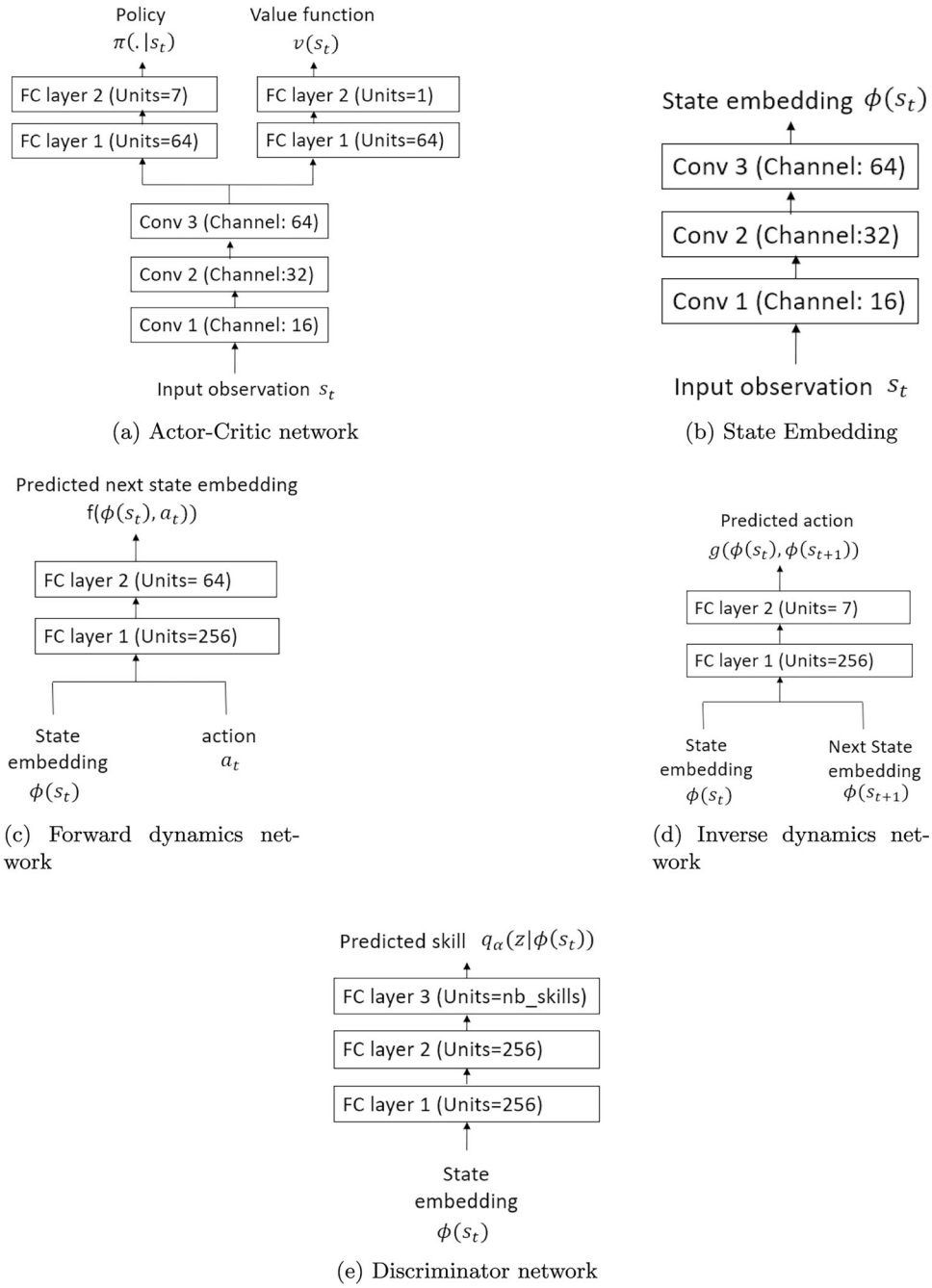


Fig. 2 Neural network architectures

finetuning. Training curves, averaged over five runs with different seeds, are provided for all algorithms. The simulations in this study were conducted on a high-performance computing node equipped with an NVIDIA TITAN X (Pascal) GPU featuring 12 GB of VRAM, an Intel Xeon E5-2640 v4 CPU operating at 2.40 GHz with 40 cores, and 62 GB of RAM.

4.5 Model architecture

Figure 1 illustrates the pipeline of the empirical study, depicting the sequential flow of inputs, outputs, and reward computations within the model. At each time step t, the input observation s_t (either a grid encoding or an RGB image) is processed by the PPO algorithm, which outputs an estimated policy and value function. The agent then

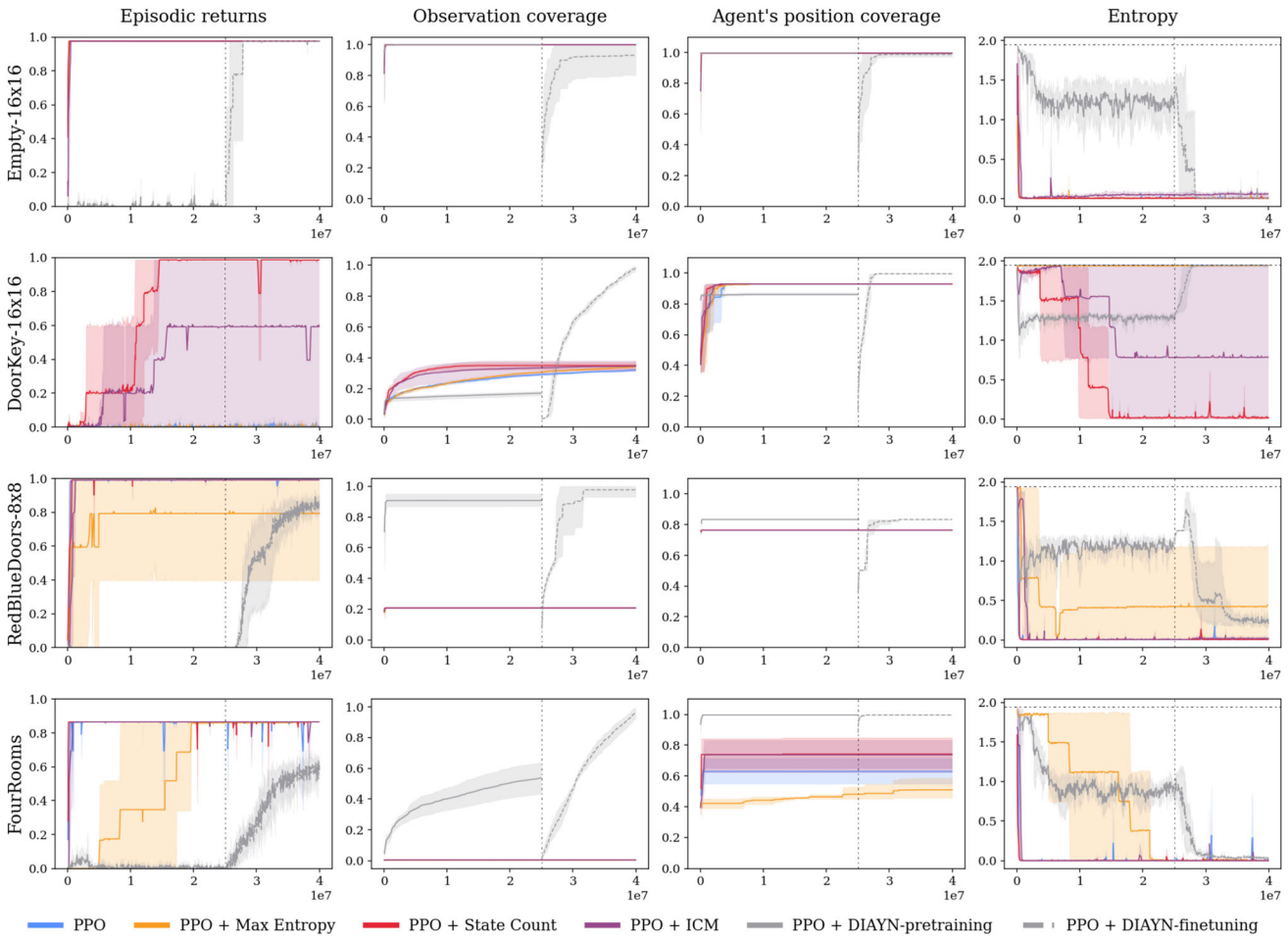


Fig. 3 Performance of four metrics—Episodic Return, Observation Coverage, Agent's Position Coverage, and Entropy—plotted against the number of transitions (frames) processed by the environment. Observations are represented as grid encodings. The results, averaged over five seeds with standard deviation shading, include evaluations across the four environments described in Sect. 4.3. The baseline model, PPO, operates without intrinsic rewards, while the other four algorithms incorporate intrinsic rewards detailed in Sect. 4.2. For DIAYN, we differentiate between pretraining (for frames $<25\,M$) and finetuning (for frames $\in [25\,M, 40\,M]$). Vertical dash-dot lines indicate the beginning of DIAYN finetuning, while horizontal dash-dot lines represent the theoretical maximum entropy of the policy, defined as $H_{max} = H(\mathcal{U}_{|A|}) = \log(|A|)$

takes an action a_t based on the estimated policy, transitions to the next state s_{t+1}, and receives an extrinsic reward $r_{ext}(t)$. Depending on the intrinsic reward method applied (State Count/SimHash, ICM, DIAYN, or Max Entropy), the agent computes an intrinsic reward $r_{int}(t)$ for the transition (s_t, a_t, s_{t+1}), following the formulations in Sect. 4.2. The intrinsic and extrinsic rewards are combined $r_{total}(t) = r_{ext}(t) + \beta * r_{int}(t)$ and fed back into the Actor-Critic PPO network to refine the policy and value function. For DIAYN exclusively, we avoid combining intrinsic and extrinsic rewards, as discussed in Sect. 4.2.

The Actor-Critic model architecture used in PPO employs a shared convolutional neural network (CNN) to process observations, which can be either grid encodings or RGB images. This CNN consists of three convolutional layers: The first layer has 16 filters of size 2×2 with ReLU activation, followed by a 2×2 max-pooling layer; the second layer has 32 filters of size 2×2 with ReLU activation; and the third layer has 64 filters of the same size and activation function. The CNN output then branches into two fully connected networks, designated as the actor and critic networks. Each network includes a hidden layer with 64 units and Tanh activation. The

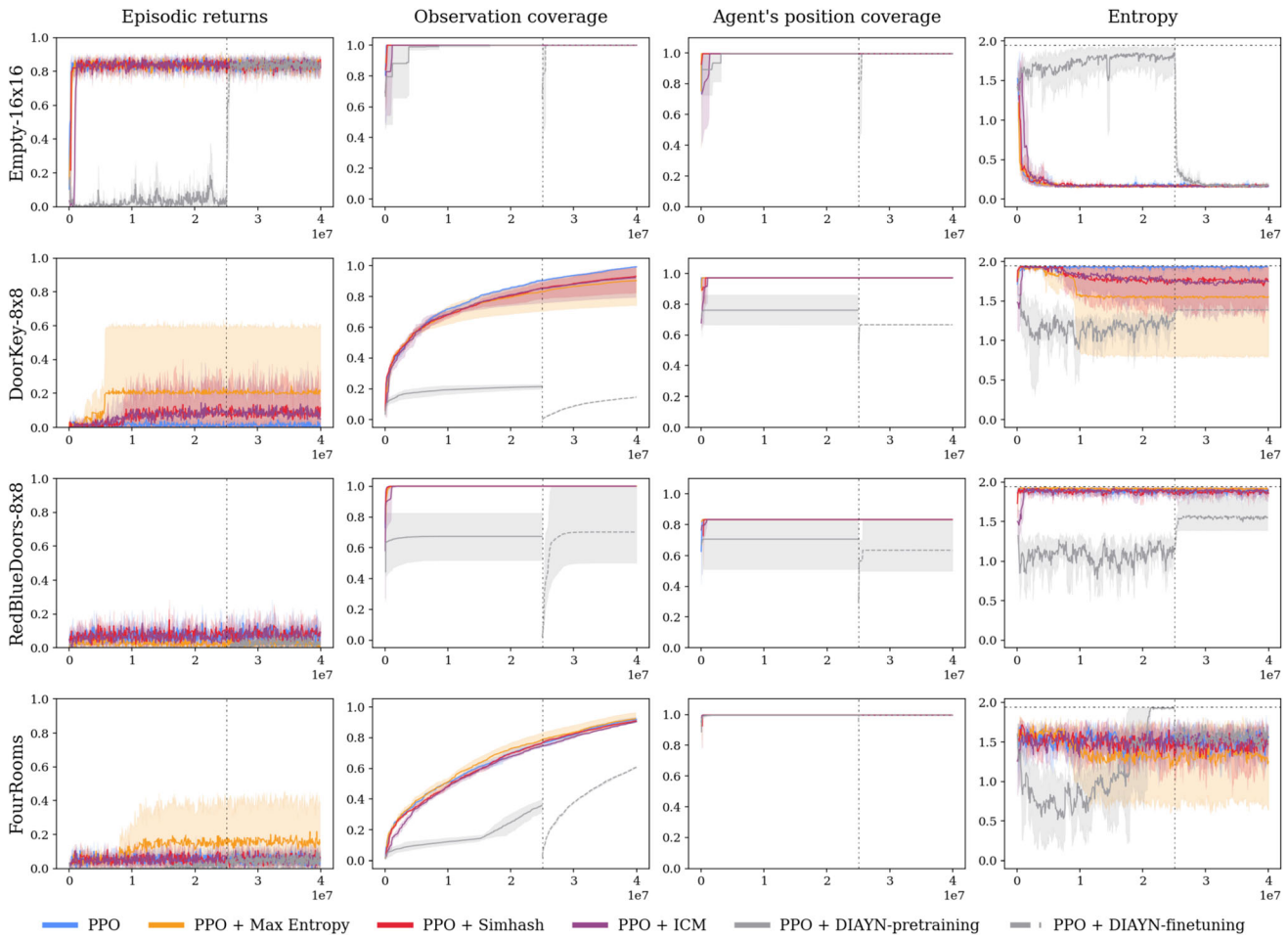


Fig. 4 Analogous to Fig. 3, but observations are partial RGB images

actor network produces action probabilities, while the critic network outputs the value function. Figure 2a provides an overview of this architecture.

The PPO architecture remains consistent across all intrinsic rewards. Some methods, however, require additional auxiliary networks, such as the embedding networks ϕ for ICM, DIAYN, and SimHash (see Fig. 2b), forward (f) and inverse (g) dynamics networks in ICM (Fig. 2c and d), and the discriminator network q_α in DIAYN (Fig. 2e). For ICM and DIAYN, the state embedding network follows the same CNN architecture as PPO to extract features from observations (see Fig. 2b). SimHash further appends a fully connected layer to the embedding network, reducing the RGB image embedding to a 512-dimensional vector prior to hashing.

4.6 Evaluation metrics

We analyze each of the intrinsic rewards, according to five metrics:

Episodic return: We plot the episodic extrinsic return averaged over the number of parallel actors $\frac{1}{N}\sum_{n=1}^{n=N}\sum_{t=1}^{t=H} r_t^n$, where N is the number of actors, and H is the length of the episode. This metric captures the convergence speed and learning ability of the intrinsic reward method.

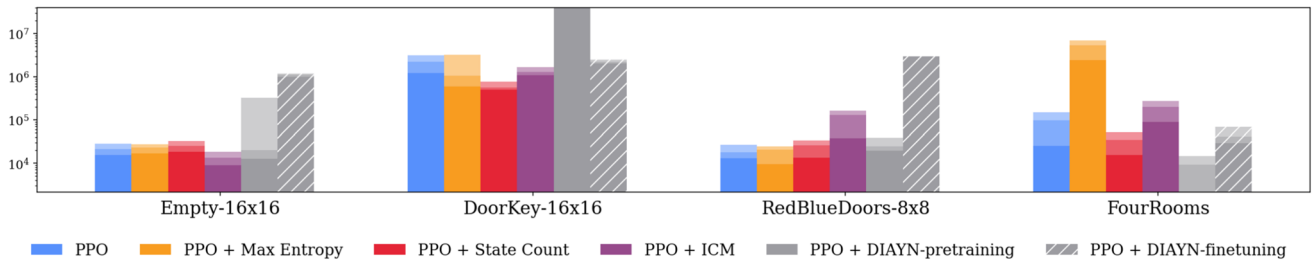


Fig. 5 Histogram showing the average number of frames required for each exploration method (PPO baseline and the four intrinsic rewards detailed in Sect. 4.2) to discover rewards across the environments described in Sect. 4.3. Observations are grid encodings. Each bar is divided into three progressively fading compartments, representing the frames at which the first, second, and third rewards are collected during training, with lower values indicating better performance. Results are averaged over five runs. For variation measures alongside average results, see the tables in Appendix D

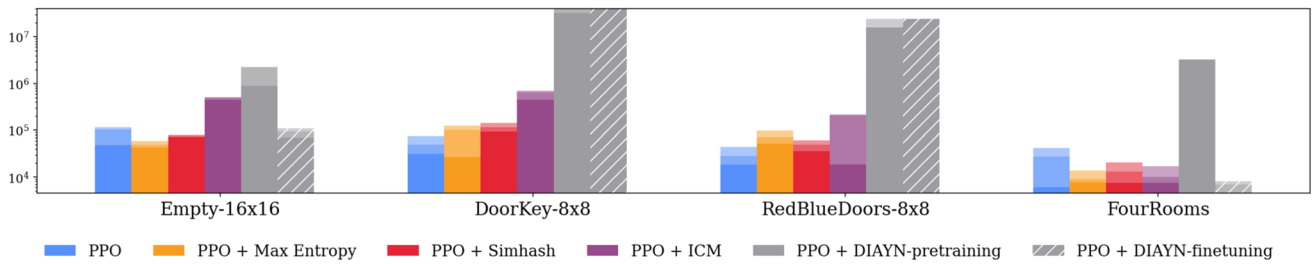


Fig. 6 Analogous to Fig. 5 but observations are partial RGB images

Observation coverage: This metric offers insight into the extent of exploring the observation space. We count how many unique observations (grid encodings or RGB) have been visited during training. We normalize this metric over the highest coverage achieved by the intrinsic reward methods. Observation coverage mirroring state coverage suggests that the neural network's embedding captures historical information and effectively represents the state.

Agent's position coverage: This metric shows the proportion of (x, y) grid positions visited by the agent so far during training: $N_{visited}(x, y)/N_{total}(x, y)$. $N_{total}(x, y)$ is the number of all possible grid positions that the agent can visit. This metric captures how well the agent has explored the position space, which is different from the observation space in a partially observable framework.

Policy Entropy: This metric evaluates the level of stochasticity of the policy. We use the average Shannon entropy: $H = -\frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{7} p(a_j)\log p(a_j)$ where T is the number of frames.

Time steps of the first, second, and third reward discoveries: We record the number of frames at which the sparse reward is found by each intrinsic reward for the first, second and third times. Note that "number of frames" refers to the number of times the agent interacted with the environment throughout the training. This metric sheds light on the speed and effectiveness of the exploration method to discover the high-reward states, as well as learning to revisit these states.

Finally, we include further visualizations (heatmaps) of the state visitation count ((x, y) positions) in Appendix D. These heatmaps represent the proportion of visits to each grid position (x, y) relative to the total number of frames. To generate them, we train the agent for $10M$ frames in singleton environments, where the maze layout remains fixed across training episodes. This setup highlights the agent's exploration patterns on a consistent grid map. Figs. 9, 10, 11, and 12 in Appendix D.1 display results for grid-encoded observations, while Figs. 13, 14, 15, and 16 in Appendix D.2 show results for RGB observations. These visualizations illustrate the

areas of the grid explored by the agent during training across the four environments: Empty, DoorKey, FourRooms, and RedBlueDoors.

5 Experimental results and discussion

We discuss the following three questions to analyze the performance of the exploration algorithms:

- RQ1: Do different intrinsic rewards lead to different return performance/sample efficiency for both grid encodings and RGB partial observations?
- RQ2: What are the characteristics (strengths/weaknesses) of each intrinsic reward method, and what are the practical recommendations to select intrinsic rewards?
- RQ3: How do different intrinsic rewards impact efficiency in discovering the sparse reward? Is there any link with credit assignment?

5.1 RQ1: Return performance of the different intrinsic rewards

In terms of episodic return, State Count has the best performance with low-dimensional observations (grid encodings) on all environments (see column 1 of Fig. 3). It converges to the maximum return with the least number of frames. In the case of DoorKey 16x16, where many algorithms—including PPO, Max Entropy, and DIAYN—struggle to solve the task, State Count emerges as the top performer, successfully obtaining the key and attaining the highest return. Following closely, ICM demonstrates lower sample efficiency. However, this is not the case for RGB observations (refer to column 1 of Fig. 4), in which SimHash (equivalent to State Count) performs poorly on most environments. The failure of SimHash in the case of RGB observations can be attributed to the challenge in adequately representing the significant features present in the high-dimensional states. RGB images contain an abundance of extraneous pixel-level details that are irrelevant to the task, requiring the agent to represent only the meaningful features. SimHash, which uses a simple hashing mechanism to represent states, struggles to capture the essential features in RGB states due to their sparse and coarse encoding mechanisms. This limitation is especially evident in environments that require high level of feature abstraction and attention to object relationships, such as DoorKey 16x16, where misrepresenting critical details hinders the agent's navigation.

Max Entropy is less impacted by such representation learning difficulties. It achieves a slightly higher return on DoorKey 8x8 and FourRooms environments in the case of RGB observations (Fig. 4). This robustness can be attributed to Max Entropy's tendency to encourage diverse policy exploration without heavily relying on specific state representations, which provides a certain level of resilience to noisy feature extraction. All other intrinsic rewards struggle to solve the tasks (except for Empty 16x16) and consistently maintain a high level of nondecreasing policy entropy. This is likely because these methods rely on high-quality state representations to produce meaningful novelty signals. In high-dimensional RGB observations, however, they tend to generate less informative intrinsic rewards. This results in difficulty differentiating between truly novel states and irrelevant pixel-level variations, causing policy learning to stagnate.

DIAYN finetuning has a worse average return compared to the baseline PPO in both grid encodings and RGB scenarios. This shows that initializing the Actor-Critic weights with DIAYN skills does not improve sample efficiency compared to random initialization. Note that DIAYN pretraining does not collect any extrinsic reward because it is trained to maximize the intrinsic reward generated by the discriminator and not the true task reward. We hypothesize that the limited skill label space, compared to the vast state space, promotes the learning of static skills that lack adaptability and fail to transfer effectively to the target task. Specifically, the states encountered by different skills tend to vary only slightly, enabling skill differentiation but not necessarily the development of semantically meaningful or broadly transferable skills.

5.2 RQ2: Characteristics of each intrinsic reward algorithm

State Count/SimHash demonstrates the best sample efficiency in grid encodings, enabling efficient task-solving in small state/action spaces. Additionally, it ensures a fast coverage of observations and grid positions compared to other algorithms, as depicted in columns 2 and 3 of Figs. 3 and 4. Furthermore, as it converges to the optimal policy, it exhibits a fast decreasing policy entropy due to the diminishing intrinsic reward effect with increasing state counts. Examination of the heatmaps (Appendix D) reveals that State Count offers the most uniform coverage of the state space across all environments. This enables the algorithm to identify the optimal path, while maintaining a balanced approach between exploration and reward maximization. Remarkably, in the DoorKey environment (Fig. 10 in Appendix D), State Count demonstrates a tendency to revisit the area around the key more frequently. However, despite these strengths, a notable limitation arises in its inability to effectively handle RGB images. In such cases, the algorithm struggles to accurately count or represent pixels, thereby limiting its applicability in scenarios with high-dimensional state spaces. As a practical recommendation, State Count is a good choice for small, discrete environments, but struggles with complex, high-dimensional ones. Although we did not explore how different representations impact the performance of State Count in this study, incorporating representation learning techniques presents an interesting avenue for future research.

Intrinsic Curiosity Module (ICM) exhibits favorable return performance and effectively explores the observation and position spaces, similarly to State Count, as they both prioritize exploration within the state space. In environments characterized by low-dimensional state spaces (Fig. 3), ICM showcases consistent stability in solving tasks across diverse scenarios. However, ICM's convergence speed generally lags behind State Count due to the added computational complexity of training both forward and inverse dynamics models. This additional overhead likely introduces inefficiencies that slow down exploration, as shown in heatmap analyses (Appendix D), where ICM's slower rate of grid position exploration is evident. These heatmaps illustrate that while ICM achieves thorough state coverage, it does so at a slower rate, potentially limiting its efficiency in tasks requiring rapid convergence. Moreover, similarly to State Count, ICM encounters challenges in effectively processing RGB images (Fig. 4). The pixel-based inputs add significant complexity, making it difficult for ICM's dynamics models to effectively process and encode meaningful features. This limitation suggests that ICM's performance may be hampered in visually complex environments.

Max Entropy can solve most environments in the case of grid encoding observations (except DoorKey) (Fig. 3). However, it does not converge faster than State Count and shows a slightly lower average return because it fails for some of the runs (such as on RedBlueDoors). This instability arises from the algorithm's tendency to promote high stochasticity in the policy, even when a more deterministic approach would suffice, ultimately affecting the average performance. By analyzing the heatmaps, we can see that Max Entropy explored unnecessarily or became confined to certain regions of the state space, especially in easy environments such as FourRooms and RedBlueDoors (Figs. 11 and 12 in Appendix D). This unnecessary exploration delays convergence to optimal paths, as the agent is distracted from effectively reaching the goal. The algorithm's inclination to prompt the agent to try all possible actions, including those rarely relevant to task success, can divert focus and hinder progress. Additionally, a drawback of the Max Entropy approach is that states with lower entropy may be visited less frequently or even overlooked. As discussed by Han and Sung [55], the maximum entropy strategy, which optimizes policies to reach high-entropy states, does not always foster effective exploration. Rather, it can create positive feedback loops where the agent becomes overly focused on high-entropy areas, limiting its ability to comprehensively explore the environment. This might reduce the likelihood of reaching less-visited yet potentially critical states.

Nevertheless, in the case of partial RGB observations (Fig. 4), Max Entropy is less impacted by representation learning challenges. We observe that on DoorKey and FourRooms, it slightly outperforms SimHash in terms of return and shows a decrease in the policy entropy (see columns 1 and 4 of Fig. 4), as it succeeds in reaching the goal in several runs. Therefore, for grid-based settings with high-dimensional state spaces, where simply counting states becomes impractical, maximizing entropy can be a valuable alternative exploration strategy to State Count.

As a practical recommendation, Max Entropy may not be the most effective exploration method in grid-like environments with high-dimensional action spaces, where many actions are unused. However, it performs adequately in environments with large state spaces and small action spaces.

DIAYN generally has the worst average return compared to the other three intrinsic rewards in both grid encodings and RGB scenarios. This is attributed to the tradeoff between the ability to discriminate between different skills and optimality. The need to generate distinguishable skills often leads DIAYN to prioritize visits to easily discriminable states over achieving optimal exploration. In the case of low-dimensional state space (Fig. 3), it is surprising that DIAYN finetuning has the highest observation and position coverages on most environments (DoorKey, FourRooms and RedBlueDoors). The ease of discriminating observations (due to distinct grid encodings reflecting different object types, colors, or status) drives DIAYN to prioritize visiting them. Unlike environments with distinct features, DIAYN struggles to cover the observation space in Empty 16x16 due to the difficulty of discriminating observations in a near-uniform grid (mostly walls). This further underscores DIAYN's reliance on environments with clear, discriminable features for effective exploration. Moreover, the poor state space coverage by DIAYN (both pretrained and finetuned) in the RGB setting (Fig. 4) indicates limitations in the discriminator's ability to discriminate between RGB observations. This suggests that the additional challenge of representation learning exacerbates the discriminator's learning difficulties. The presence of high-dimensional visual data introduces an added layer of complexity as the agent must learn both to distinguish visual features and navigate the space effectively. By further analyzing the exploration pattern of DIAYN through the heatmaps, we notice the following: DIAYN demonstrates uneven state coverage, often focusing on corner areas or becoming restricted to specific regions within the grid that contain easily distinguishable states (For example, see Figs. 10, 12, 15, and 16 in Appendix D). This suggests potential limitations in its ability to explore diverse regions and acquire transferable skills. Without reaching different target positions (e.g., door/key/goal), the skills lack meaningful variations and adaptability. We hypothesize that this is due to DIAYN's mutual information (MI) objective, which does not explicitly maximize the entropy of the state distribution [56] and does not promote broad state coverage [57]. The agent tends to receive higher rewards for visiting known states rather than exploring novel ones, as fully discriminable states yield a high MI reward [58]. This can hinder novel state exploration and discourage the agent from learning far-reaching skills [56]. Consequently, DIAYN might potentially constrain the diversity of learned skills to those that are easier to distinguish but not necessarily effective for broad exploration or task relevance. In our particular setting, DIAYN also encounters difficulty in learning the abstract skill space effectively. This challenge might be particularly pronounced due to partial observability. As a practical recommendation, learning unsupervised skills with DIAYN does not help exploration in MiniGrid framework, especially in strategic tasks. Nevertheless, pushing for diversity of skills can be useful for skill-chaining, fast adaptation to environment changes, robustness, and generalization to different tasks. We emphasize that our results hold only for our particular setting where skill-learning turns out to be antagonistic to exploration and sample efficiency in MiniGrid. This might not hold in other environments that could benefit from such skills to converge faster. It is also worth noting that exploring factors such as the skill space, the initial skill distribution, and incorporating state abstraction techniques, along with auxiliary exploration mechanisms to enhance state coverage of skills, could significantly improve DIAYN's performance. However, because this constitutes a substantial variation from the original algorithm, we leave these considerations for future work.

5.3 RQ3: First, second, and third instances of discovering the sparse reward

We record the timesteps at which the sparse reward is found by each of the intrinsic rewards for the first, second, and third times in both grid encodings (Fig. 5) and RGB (Fig. 6) scenarios. For more detailed results, including averages and standard deviations, refer to Tables 4, 5, 6, and 7 in Appendix D.1, as well as Tables 8, 9, 10, and 11 in Appendix D.2. We notice that in the case of low-dimensional observation space, State Count (which has the highest return performance) finds the reward soon on most environments, while DIAYN takes time to reach the goal, especially on strategic tasks. For example, in the DoorKey environment, which represents a strategic task,

State Count is the first intrinsic reward to find the task reward, while DIAYN finetuning is the last, and DIAYN pretraining does not reach the goal at all within the pretraining time. This shows that DIAYN exhibits limitations in acquiring skills that achieve the goal sequence of visiting the key, the yellow door, and the green goal in this specified order. This limitation is likely due to DIAYN's focus on skill diversity rather than directed exploration, making it less effective in tasks requiring structured sequences. Another interesting observation is that the algorithm that discovers the reward the fastest for the first time, might not be the fastest in visiting the rewarding state a second and third time. This implies that diversity impacts credit assignment. For example, on DoorKey (see Table 5 in Appendix D), Max Entropy finds the first reward before ICM for the first time, but it takes more time to learn that it should go back to the reward for the third time. This is paramount to designing a sample efficient algorithm because visiting rewards more often provides more informative learning signals and allows learning credit faster, more accurately and with less variance [59]. This implies that although Max Entropy promotes policy exploration, it may lack mechanisms for prioritizing or remembering rewarding states that consistently provide useful learning signals.

In contrast, the results vary across other environments, highlighting how different algorithms perform under varying conditions. Notably, in the FourRooms environment (Fig. 5), DIAYN pretraining is the first to find the reward, as opposed to the case of strategic tasks. In an environment consisting of several identical compartments, learning skills could lead to quick reward discovery even though it does not directly maximize the task reward. This suggests that DIAYN might be advantageous in environments with structural similarity, where learned skills can be reused across similar compartments. For the case of RGB observations (Fig. 6), we observe that PPO and Max Entropy are among the fastest methods to find the reward on most environments, surpassing SimHash. This reinforces the hypothesis that, when scaling to high observation spaces, entropy might be a better strategy to push for exploration rather than counting states. DIAYN finetuning also takes a long time to find the reward, especially for strategic tasks such as DoorKey and RedBlueDoors. This suggests that DIAYN's emphasis on diversity may limit its ability to prioritize reaching task-relevant states in complex, sequential tasks. Integrating the mutual information (MI) objective of DIAYN with trajectory-based metrics between states to enhance exploration could be a potential direction for handling strategic tasks.

6 Conclusions

In this work, we have reinterpreted intrinsic reward techniques in the literature using a diversity perspective (State, State + Dynamics, Policy, and Skill levels of diversity). We conducted empirical studies on MiniGrid, to understand the differences between these diversity levels in a partially observable and procedurally generated framework.

We found that the homogeneity of the state coverage imposed by State Count (representing State level diversity) has led to the best sample efficiency on many MiniGrid tasks. State level diversity improves the convergence speed in strategic tasks, covers the state space well and leads to a fast decrease of policy entropy and intrinsic reward. This decreasing rate of the intrinsic reward aligns well with finding the optimal policy which avoided the dominance of the intrinsic reward. However, State level diversity is fragile and requires good state representations, while entropy maximization seems to be slightly more robust when dealing with image-based observations. Learning good state representations is challenging, so entropy maximization is a practical alternative. Moreover, DIAYN (representing Skill level diversity) struggles with exploration in MiniGrid due to the difficulty of learning the skill space and exploring within it, in a procedurally generated partially observable setting.

6.1 Limitations and future works

This study serves as an initial exploration into the relationship between exploration and diversity imposed by intrinsic rewards. While we provide insights into this relationship, several limitations remain to be addressed in the future work.

Firstly, we examine only one representative intrinsic reward method for each level of diversity. This choice may not capture the full range of behaviors within each category, potentially limiting the generalizability of our findings. Expanding this work to benchmark a broader selection of intrinsic reward methods would improve the applicability of our results.

Additionally, the effectiveness of intrinsic rewards is closely tied to the environment in which they are applied. Our experiments are restricted to the MiniGrid environment, specifically using grid encodings and RGB observations. Future studies could benefit from exploring more complex and varied environments, such as Mujoco [60], Atari [37], MiniHack [45], and MiniMax (Autocurricula) [61], where the impacts of different diversity levels might yield more distinct behaviors. Some intrinsic reward methods may excel in certain environments but perform poorly in others. Thus, identifying conditions under which each intrinsic reward method performs best across diverse environments would be a valuable contribution.

Moreover, while diversity can enhance exploration, it may also impede performance as discussed in [62] in a phenomenon named *the curse of diversity*. Therefore, pinpointing the conditions under which diversity aids rather than hinders performance-or developing strategies to counterbalance the potential negative effects of diversity-remains an open research question.

For the *competence-based* category, we employed DIAYN, a method that learns a skill space autonomously. Other goal-conditioned approaches, such as those learning different goal representations [63], predefining goal abstractions [64], or employing careful skill composition/chaining, may yield more efficient exploration strategies. Investigating these approaches could offer further insights into *competence-based* intrinsic rewards.

Finally, representation learning—especially as applied in conjunction with intrinsic reward methods—also significantly impacts exploration efficacy. Analyzing how representation learning interacts with different levels of diversity and affects exploration performance is an important direction for future research.

Another interesting future work could explore integrating intrinsic reward diversity with context-aware RL [65–67]. This integration could improve exploration strategies by adapting them to specific environmental or task contexts. However, this is beyond the scope of the current work, which focuses solely on intrinsic rewards without additional adaptive mechanisms.

Appendix A: Diversity levels categorization

We divide intrinsic rewards into two categories: “Where to explore” and “How to explore?”, as described in the following and shown in Fig. 7.

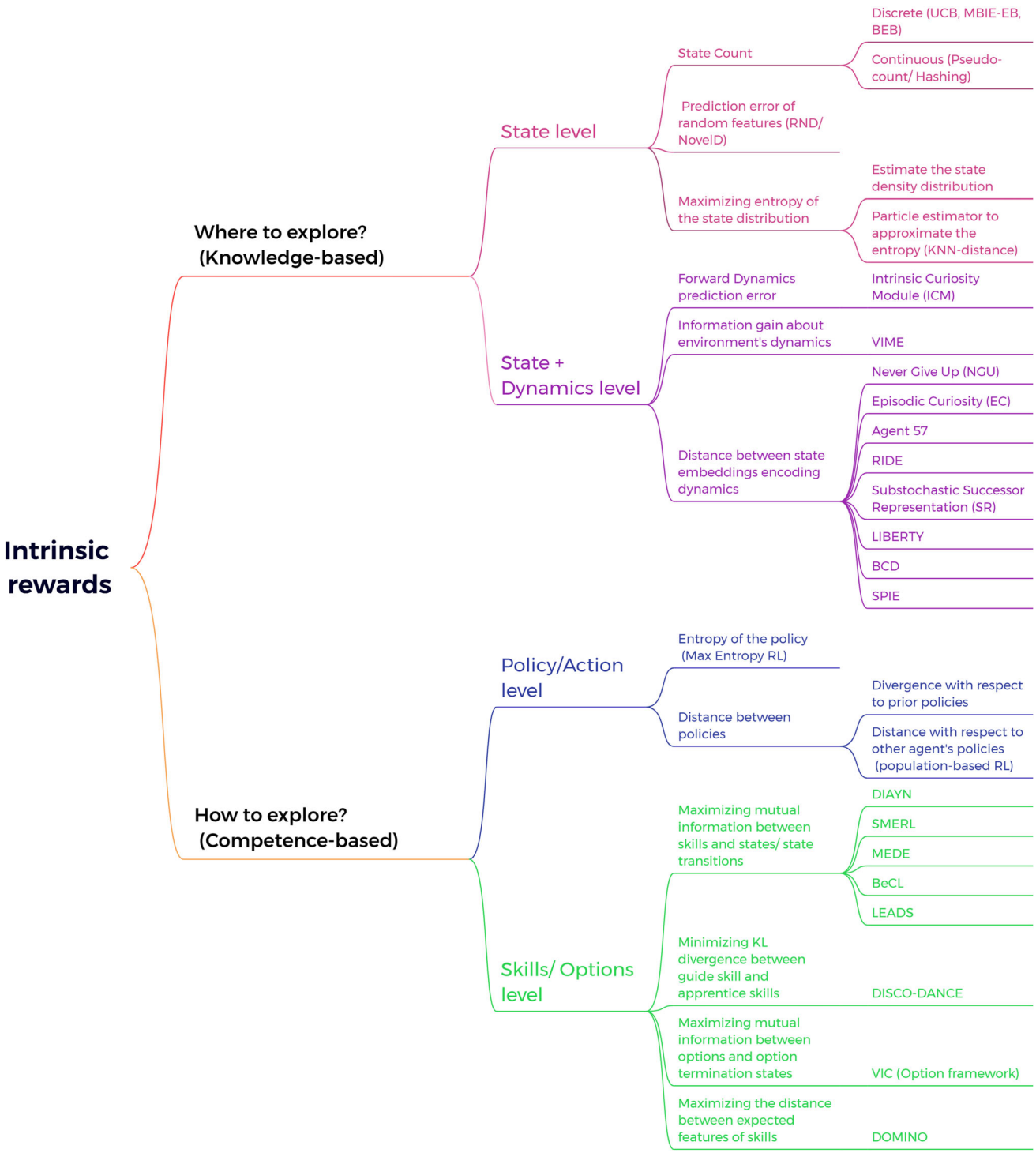


Fig. 7 Categorization of the different levels of diversity incurred by intrinsic rewards for exploration in RL

Appendix A.1. *"Where to explore?"*

State level diversity In this subcategory, we collect all the works that encourage the exploration of unseen states. 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 [33, 68, 69]. 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 [13] and hashing [14].

Besides count-based methods, feature prediction error can be used as a measure of state novelty. For example, in [15], the 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 [70] uses RND as a measure of state novelty but 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 that aim to maximize the entropy of the state distribution induced by the policy over a finite or infinite horizon by estimating the state density distribution [71, 72] or by relying on the K-nearest neighbor (KNN) distance as an approximation of state entropy [40, 73–75].

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 that use the forward dynamics prediction error as an intrinsic reward, such as [17] and [76]. The key intuition is to encourage the agent to revisit unfamiliar state transitions where the prediction error is high (high mismatch between the agent's expectation and true experience). Another curiosity-driven technique is Variational Information Maximizing Exploration (VIME) [18], which pushes the agent to explore states that lead to a larger change in the dynamics model (higher information gain).

Moreover, this subcategory includes techniques that estimate the state novelty within a feature space designed to capture the temporal or dynamical aspects of states. For instance, Exploration via Elliptical Episodic Bonuses (E3B) [41] and RIDE [34] both utilize an inverse dynamics model (ICM) to learn state embeddings that represent the controllable dynamics of the environment. While RIDE encourages the agent to select actions that produce substantial changes in the state embedding, E3B applies an elliptical episodic bonus to guide exploration. Additional examples include Never Give Up (NGU) [16], Agent 57 [77], and Episodic Curiosity (EC) [78], all of which employ memory-based methods using distance-based metrics in a dynamics-aware feature space to approximate State + Dynamics novelty. Similarly, [79] propose the LIBERTY approach, which utilizes an inverse dynamic bisimulation metric to measure distances between states in a latent space, ensuring effective exploration and policy invariance. The work of [80] also presents a novel behavioral metric with Cyclic Dynamics (BCD), leveraging successor features and vector quantization to evaluate behavioral similarity between states and capture interrelations among environmental dynamics. Finally, [81] propose using the inverse of the norm of the successor representation (SR) as an intrinsic reward to account for transition dynamics. More recently, [82] developed the SPIE approach, which constructs an intrinsic reward by integrating both prospective and retrospective information from previous trajectories, also based on SR.

Appendix A.2. *"How to explore?"*

Policy/Action level diversity Algorithms in this subcategory aim to explore diverse actions from the same state. What makes it different from the State + Dynamics algorithms introduced in Appendix A.1 is that the previous category uses knowledge about the 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 in the policy learned. For example, in Maximum Entropy RL (Max Entropy), 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 [83]. Soft Actor-Critic (SAC) [49] is a popular RL algorithm

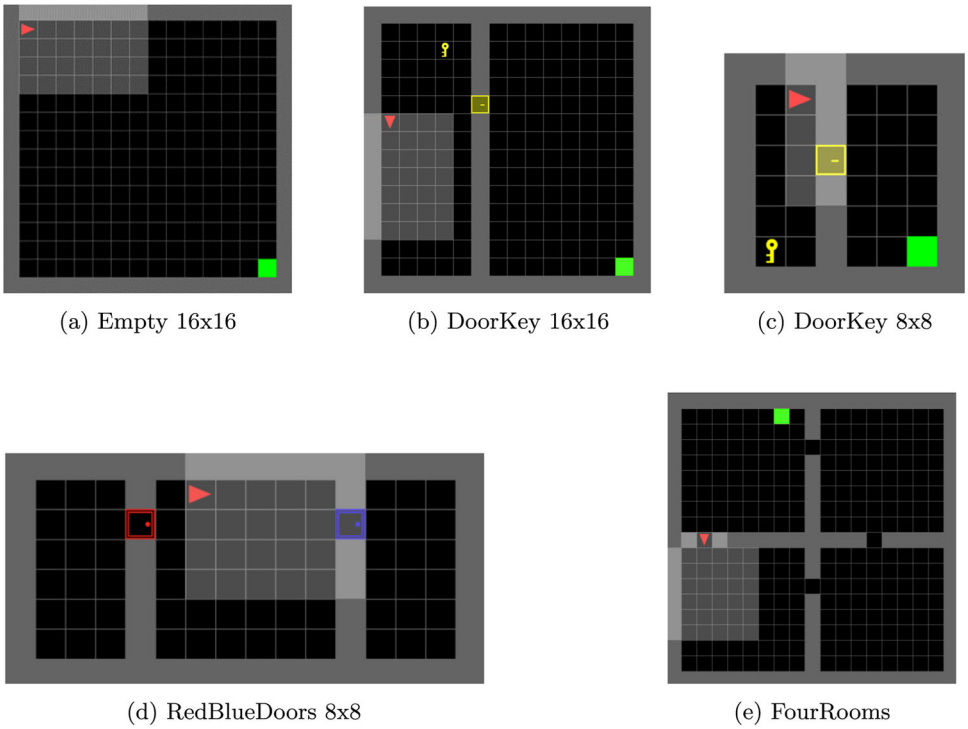
implementing the Max Entropy RL framework. Diversity-driven exploration strategy [84] is another exploration technique that encourages the agent to behave differently in similar states. It maximizes the divergence between the current policy and prior policies. Similarly, Adversarially Guided Actor-Critic (AGAC) [85] 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 behaviors by remaining different from the adversary. Another branch that belongs to this diversity level is population-based exploration, which combines evolutionary strategies with Reinforcement Learning. These approaches train a population of agents to learn diverse behaviors that are high scoring at the same time, in order to effectively explore the environment [86, 87]. For more details on the connection between evolutionary approaches and RL, please refer to the comprehensive survey by [88].

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 $\pi(a|s,z)$ [21]. This category aims to discover diverse skills, and the intrinsic reward is a function of the skill. Most methods falling into this category come from the domain of unsupervised skill discovery and use a discriminator-based architecture, such as Diversity is All You Need (DIAYN) [20]. DIAYN replaces the task reward with a learned discriminator term $q_\alpha(z|s)$ that infers the behavior from the current state in order to generate diverse policies visiting different sets of states. It also uses the Max Entropy RL framework to learn skills that are as random as possible [20]. Maximum Entropy Diverse Exploration (MEDE) [21] is very similar to "DIAYN + extrinsic reward", with the small difference of conditioning the discriminator on the state-action pair $q_\alpha(z|s,a)$ instead of the state only. Moreover, MEDE uses the discriminator term as a prior in the objective function instead of adding it as an intrinsic reward. Structured Max Entropy 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 returns [25]. 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 [26]. It is important to mention that skills in the literature can be called options or goals. For example, Variational Intrinsic Control (VIC) is an algorithm that provides the agent with an intrinsic reward that relies on modeling options and learning policies conditioned on these options [22]. Instead of sampling options from a fixed prior distribution as in DIAYN, VIC learns the prior distribution of options and updates it to choose options that yield higher rewards [22]. Both DIAYN and VIC are part of goal-conditioned RL methods, where goals are internally generated by agents and achieved via self-generated rewards [11]. More recent unsupervised skill-learning methods have emerged, such as [56] which proposed Behavior Contrastive Learning (BeCL), a novel *competence-based* method that uses contrastive learning to encourage similar behaviors within the same skill and diverse behaviors across different skills. This is done by maximizing the mutual information between different states generated by the same skill as an intrinsic reward. Another recent work by Kim et al. [58] proposed skill discovery with guidance (DISCO-DANCE) which identifies the guide skill most likely to reach unexplored states, directs other skills to follow it, and disperses them to maximize distinctiveness. Moreover, Tolguenec et al. [57] proposed LEADS, which maximizes a variant of the mutual information between skills and states, by leveraging the successor state measure to tailor skills toward less-visited states while also maximizing the disparity between skills.

Appendix B: MiniGrid environments

We use the following MiniGrid environments shown in Fig. 8:

1. Empty: This is an empty grid where the agent is always placed in the corner opposite to the goal. The task is to get to the green goal square. We use the regular variant "MiniGrid-Empty-16x16-v0."



(a) Empty 16x16 (b) DoorKey 16x16 (c) DoorKey 8x8

(d) RedBlueDoors 8x8 (e) FourRooms

Fig. 8 MiniGrid environments

2. DoorKey: This is a sparse reward environment that requires a specific order of visiting the states to solve the task; the agent needs to pick up the key, open the door, and then get to the green goal square. It does not receive any reward after picking up the key or unlocking the door; it is rewarded only at the end of the task. We use "MiniGrid-DoorKey-16x16-v0" in the case of grid encodings and "MiniGrid-DoorKey-8x8-v0" in the case of RGB observations.
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.
4. 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."

For all tasks, a maximum number of steps t_{max} is assigned, to encourage the agent to solve the task as quickly as possible. When the agent succeeds after t steps, it receives a reward $r = 1 - 0.9 * t/t_{max}$ in all three environments. The episode ends when the agent collects the final reward or when the maximum number of steps is exceeded.

Observation and Action spaces

The observations are egocentric and partially observable. We first considered the grid encoding observations of size $7 \times 7 \times 3$. The first two dimensions (7×7) compose the tile set, and the last dimension encodes the object type (wall, door, $\cdots$), the object color (red, green, $\cdots$) and the object status (door open, door closed, door locked). Specifically, object type $\in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, object color $\in \{0, 1, 2, 3, 4, 5\}$, and object status $\in \{0, 1, 2\}$. Then, we used partial RGB visual observations of size $56 \times 56 \times 3$ (7 tiles of 8×8 pixels each) to increase the complexity of the task, as agents must extract features directly from the images. 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.

Appendix C: Hyperparameters

For State Count and ICM, we use the hyperparameters from the previous study [32]. Since Max Entropy + PPO and DIAYN 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 that result in the highest return during training. The chosen values of β are summarized in Table 2. For DIAYN, we choose to train 10 skills, which is the number used in the study by [89], and we use a discriminator learning rate of $3 \times e^{-4}$ following the implementation of the DIAYN paper [20] (Table 3). Note that we reused the same hyperparameters for the second part, where we tested on RGB observations.

Appendix D: Additional experimental results

Appendix D.1. Grid encoding observation space

See Tables 4, 5, 6, 7. Figures 9, 10, 11, 12.

Table 4 Frame number at which the reward is found for the first, second, and third time by each exploration method on Empty 16x16 environment with grid encoding observations

Empty 16 × 16	First reward	Second reward	Third reward
PPO	15,452 ± 7112	21,273 ± 11,539	28,304 ± 14,785
PPO + State count	18,428 ± 9119	25,340 ± 11,537	32,483 ± 12,888
PPO + Max entropy	16,841 ± 6916	22,768 ± 6736	27,318 ± 10,355
PPO + ICM	**8918± 3565**	**13,436 ± 6830**	**18,281 ± 9467**
PPO + DIAYN pretraining	12,668 ± 7030	20,076 ± 13,898	326,963 ± 662,300
PPO + DIAYN finetuning	1,001,862 ± 1,187,338	1,130,208 ± 1,082,785	1,207,600 ± 1,038,924

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported

Table 5 Frame number at which the reward is found for the first, second, and third time by each exploration method on DoorKey 16×16 environment with grid encoding observations

DoorKey 16×16	First reward	Second reward	Third reward
PPO	1,242,342 ± 529,863	2 276,508 ± 917,486	3,186,537 ± 1,616,226
PPO + State count	**496 486 ± 550,012**	**558,204 ± 548,684**	**783,075 ± 615,917**
PPO + Max entropy	594,649 ± 696,956	1,067,401 ± 743,704	3,300,668 ± 3,108,002
PPO + ICM	1,089,286 ± 734,419	1,287,632 ± 674,758	1,683,612 ± 539,173
PPO + DIAYN pretraining	40,000,000 ± 0	40,000,000 ± 0	40,000,000 ± 0
PPO + DIAYN finetuning	2,087,398 ± 449,537	2,221,756 ± 447,746	2,516,739 ± 689,340

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported. If the reward is never found, the frame number is set to the training budget (40 M)

Table 6 Frame number at which the reward is found for the first, second, and third time by each exploration method on RedBlueDoors environment with grid encoding observations

RedBlueDoors	First reward	Second reward	Third reward
PPO	13,136 ± 5647	**17,568 ± 8303**	26,553 ± 6733
PPO + State count	13,180 ± 8236	25,923 ± 11,537	33,545 ± 19,115
PPO + Max entropy	**9417 ± 2678**	20,464 ± 10,420	**24,432 ± 10,339**
PPO + ICM	37,721 ± 68,636	129,043 ± 175,507	162,060 ± 193,005
PPO + DIAYN pretraining	19,244 ± 10,004	24,611 ± 12,661	39,280 ± 25,334
PPO + DIAYN finetuning	2,992,614 ± 2,118,551	3,006,659 ± 2,114,575	3,033,043 ± 2,096,962

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported

https://doi.org/10.1007/s00521-025-11340-0

Table 7 Frame number at which the reward is found for the first, second, and third time by each exploration method on FourRooms environment with grid encoding observations

FourRooms	First reward	Second reward	Third reward
PPO	25,222 ± 32,606	97,033 ± 41,446	150,188 ± 104,821
PPO + State count	15,465 ± 9712	34,649 ± 11,090	51,820 ± 23,054
PPO + Max entropy	2,479,424 ± 5,498,212	5 327 913 ± 5,306,632	6,874,905 ± 5,056,693
PPO + ICM	89,433 ± 111,832	197,312 ± 171,435	274,883 ± 171,782
PPO + DIAYN pretraining	2089 ± 1178	9049 ± 5350	14 531 ± 9099
PPO + DIAYN finetuning	29,238 ± 27,103	41,376 ± 35,912	69,737 ± 53,656

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported

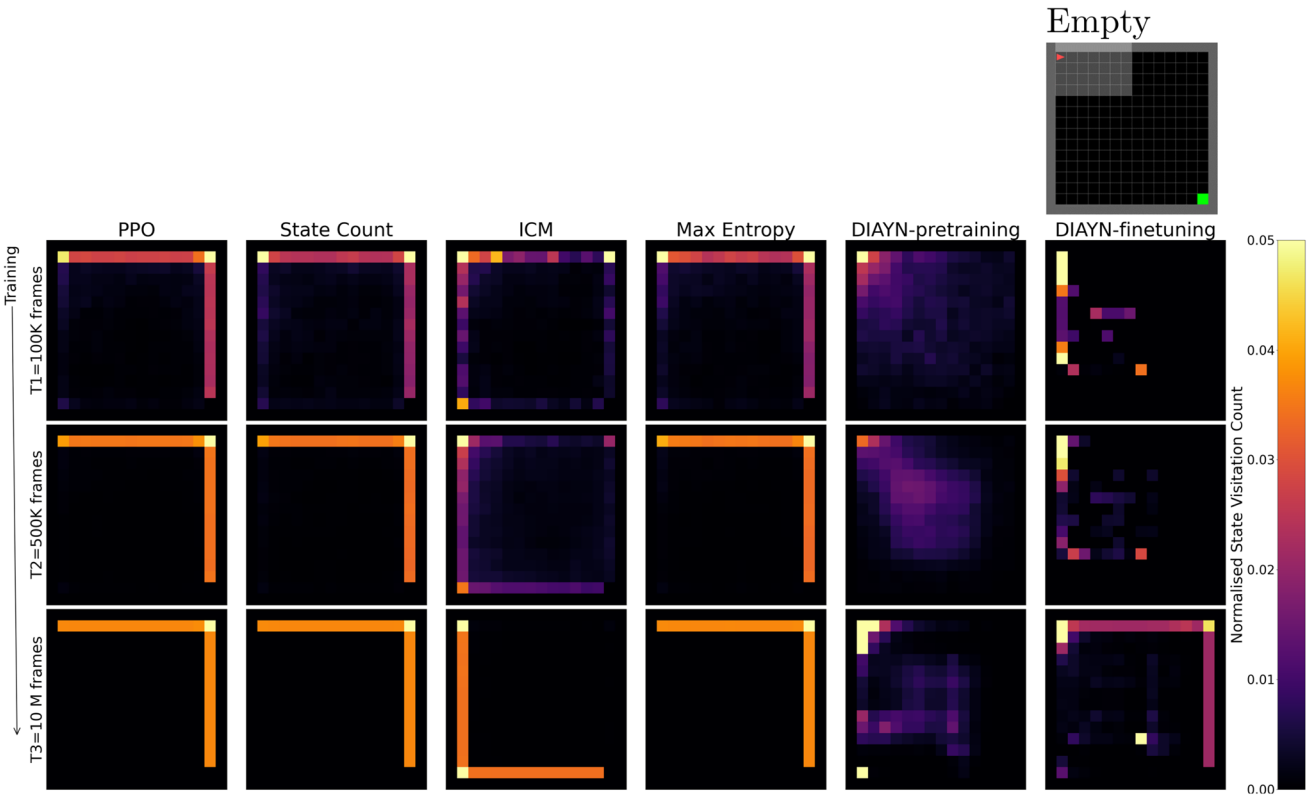


Fig. 9 State visitation count during training for 10 M frames on singleton Empty 16×16 environment with grid encoding observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames, and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

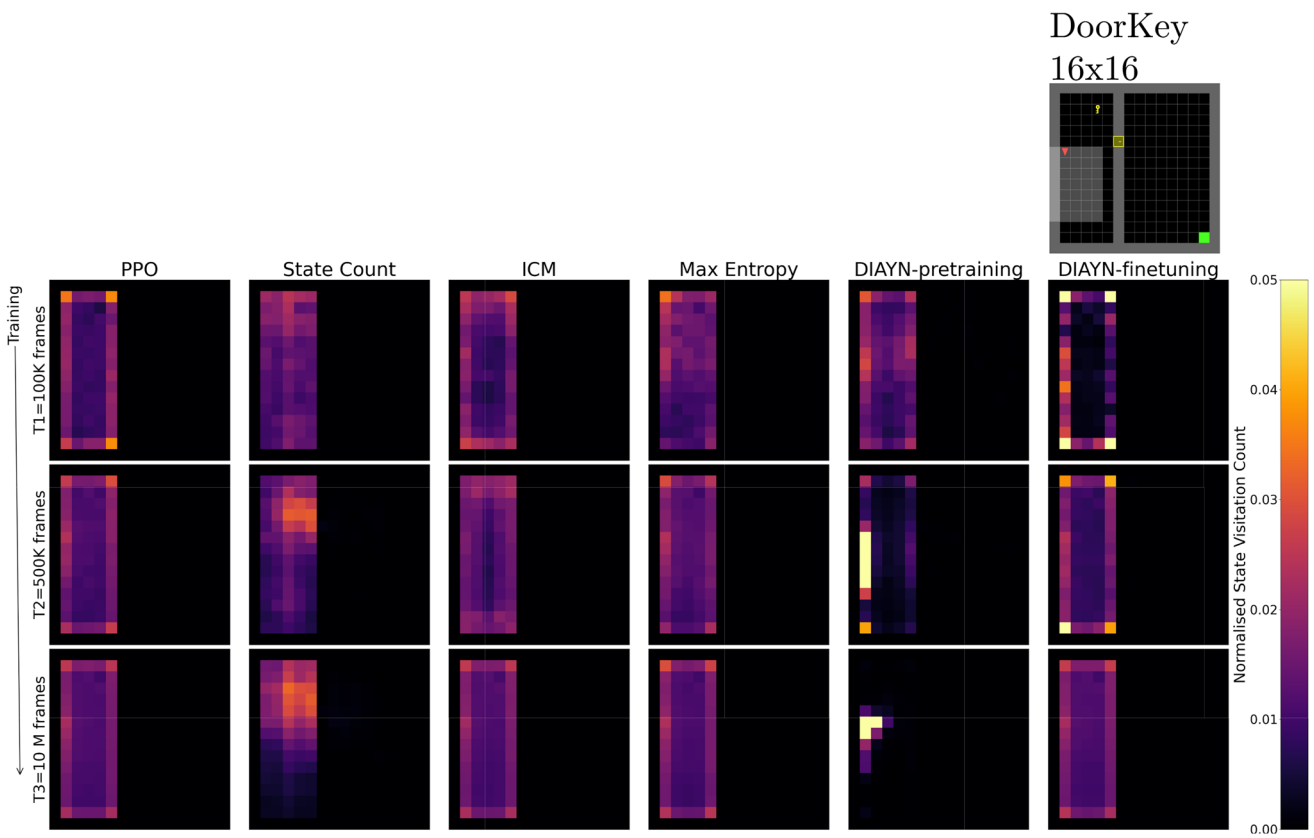


Fig. 10 State visitation count during training for 10 M frames on singleton DoorKey 16x16 environment with grid encoding observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames, and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

https://doi.org/10.1007/s00521-025-11340-0

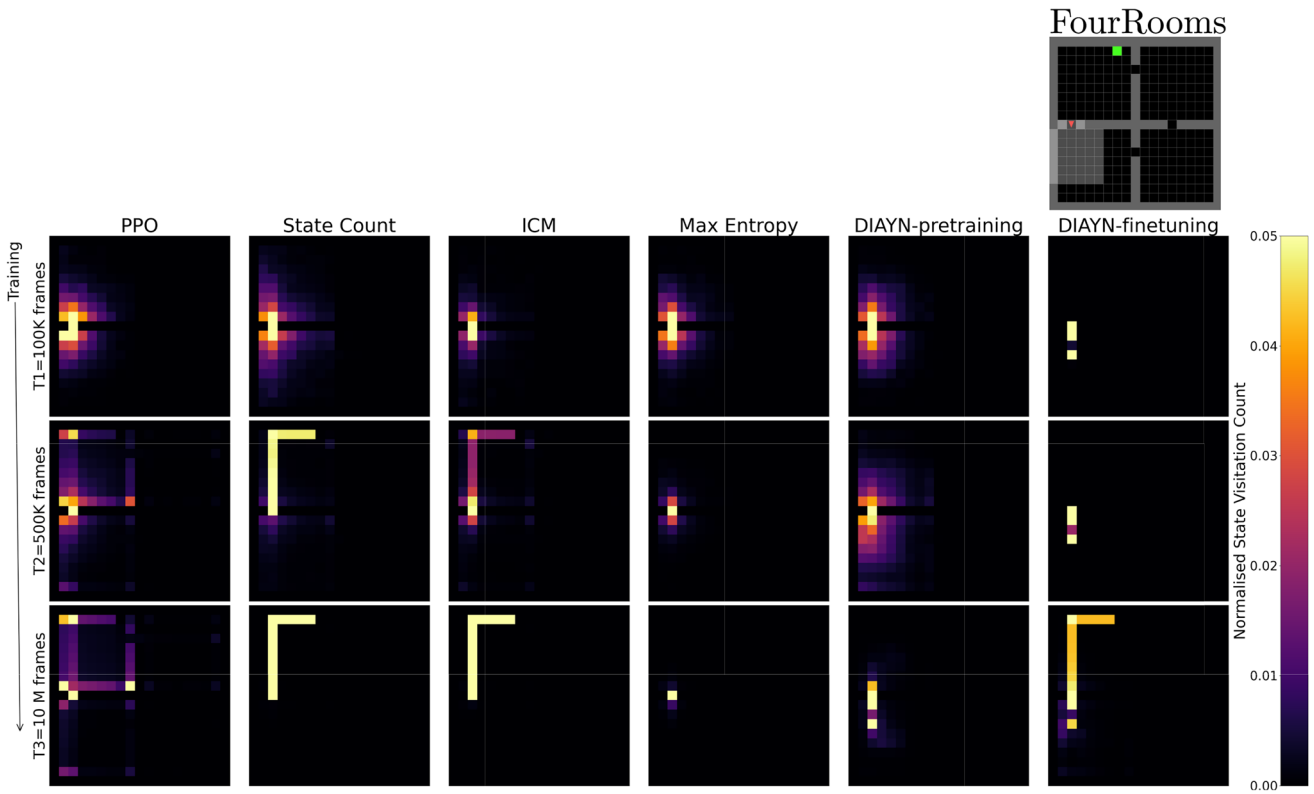


Fig. 11 State visitation count during training for 10 M frames on singleton FourRooms environment with grid encoding observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames, and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

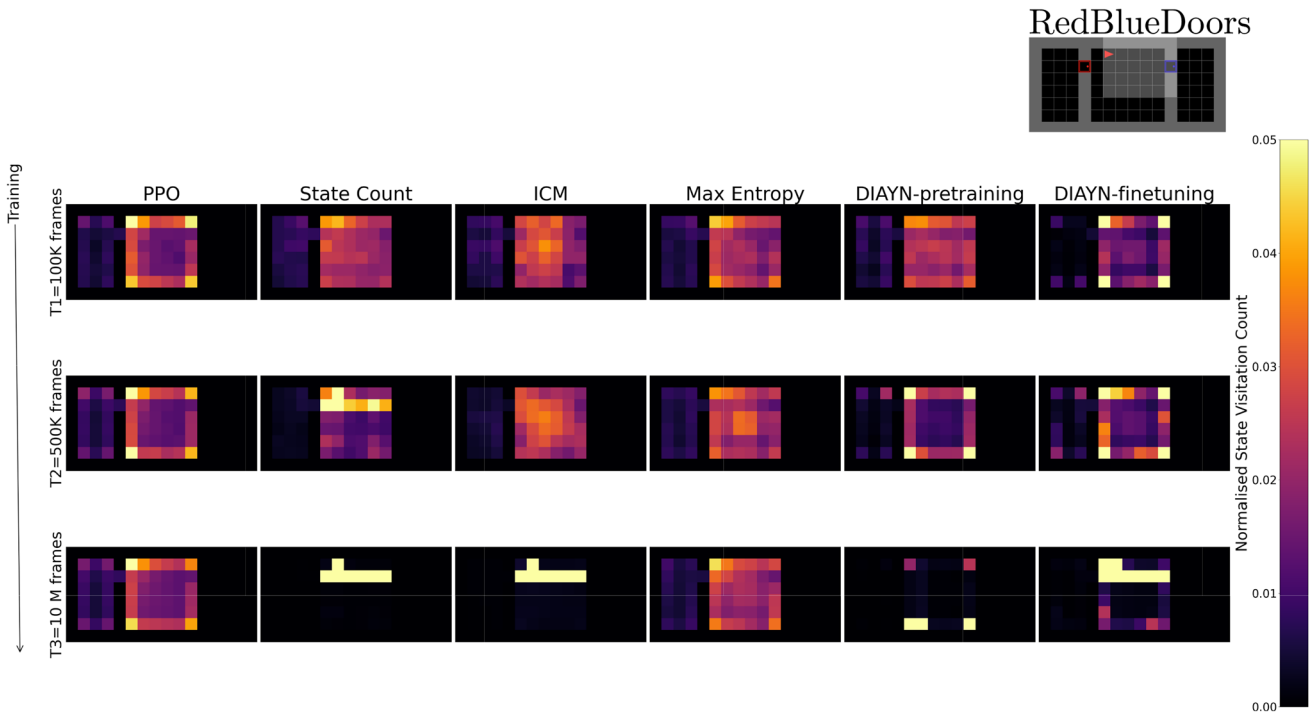


Fig. 12 State visitation count during training for 10 M frames on singleton RedBlueDoors environment with grid encoding observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames, and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

Appendix D.2. RGB observation space

See Tables 8, 9, 10, 11, Figs. 13, 14, 15, 16.

Table 8 Frame number at which the reward is found for the first, second, and third time by each exploration method on Empty 16×16 environment with RGB observations

Empty 16×16 RGB	First reward	Second reward	Third reward
PPO	48,256 ± 85,183	103,401 ± 112,583	115,148 ± 117,261
PPO + SimHash	71,132 ± 9119	75,686 ± 95,328	79,699 ± 96,431
PPO + Max entropy	**43,072 ± 76,270**	**49,033 ± 79,868**	**59,494 ± 81,361**
PPO + ICM	448,121 ± 500,641	493 401 ± 522,557	509 750 ± 509,223
PPO + DIAYN pretraining	896,963 ± 1,257,009	2 262 649 ± 3,635,427	2,267,164 ± 3,639,831
PPO + DIAYN finetuning	69,712 ± 102,945	94,240 ± 138,306	110,710 ± 135,551

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported

Table 9 Frame number at which the reward is found for the first, second, and third time by each exploration method on DoorKey 8×8 environment with RGB observations

DoorKey 8×8 RGB	First reward	Second Reward	Third reward
PPO	31,430 ± 39,987	**49,257 ± 44,984**	**75,587 ± 32,508**
PPO + SimHash	93,494 ± 75,207	115,529 ± 80,404	143,840 ± 71,717
PPO + Max entropy	**26,870 ± 34,213**	100,931 ± 96,891	124,828 ± 114,469
PPO + ICM	445,222 ± 433,991	655,200 ± 384,520	713,795 ± 381,101
PPO + DIAYN pretraining	32,003,513 ± 17,880,687	40,000,000 ± 0	40,000,000 ± 0
PPO + DIAYN finetuning	40,000,000 ± 0	40,000,000 ± 0	40,000,000 ± 0

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported. If the reward is never found, the frame number is set to the training budget (40 M)

Table 10 Frame number at which the reward is found for the first, second, and third time by each exploration method on RedBlueDoors environment with RGB observations

RedBlueDoors RGB	First reward	Second reward	Third reward
PPO	18,504 ± 12,321	**28,179 ± 19,650**	**44,342 ± 36,719**
PPO + SimHash	35,516 ± 38,700	49,548 ± 48,897	60,643 ± 58,630
PPO + Max entropy	51,871 ± 51,587	71,776 ± 59,120	97,907 ± 88,967
PPO + ICM	**18,355 ± 26,236**	206,547 ± 420,774	219,718 ± 419,780
PPO + DIAYN pretraining	16,012,892 ± 21,897,134	16,015,049 ± 21,895,167	24,011,241 ± 21,893,513
PPO + DIAYN finetuning	24,212,716 ± 21,618,947	24,212,716 ± 21,618,947	24,219,180 ± 21,610,106

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported. If the reward is never found, the frame number is set to the training budget (40 M)

Table 11 Frame number at which the reward is found for the first, second, and third time by each exploration method on FourRooms environment with RGB observations

FourRooms RGB	First reward	Second reward	Third reward
PPO	6057 ± 7369	27,203 ± 34,679	41,766 ± 47,238
PPO + SimHash	7561 ± 13,820	13,171 ± 11,599	20,406 ± 17,137
PPO + Max entropy	7654 ± 13,591	9014 ± 13,184	13,907 ± 12,435
PPO + ICM	7 491 ± 10,928	10,060 ± 12,737	16,912 ± 16,744
PPO + DIAYN pretraining	3,276,505 ± 7,322,741	3,290,252 ± 7,342,741	3,296,742 ± 7,343,739
PPO + DIAYN finetuning	**4470 ± 2384**	**6854 ± 4956**	**8 166 ± 4735**

Results are averaged over five runs. Mean and standard deviation ($\mu \pm \sigma$) are reported

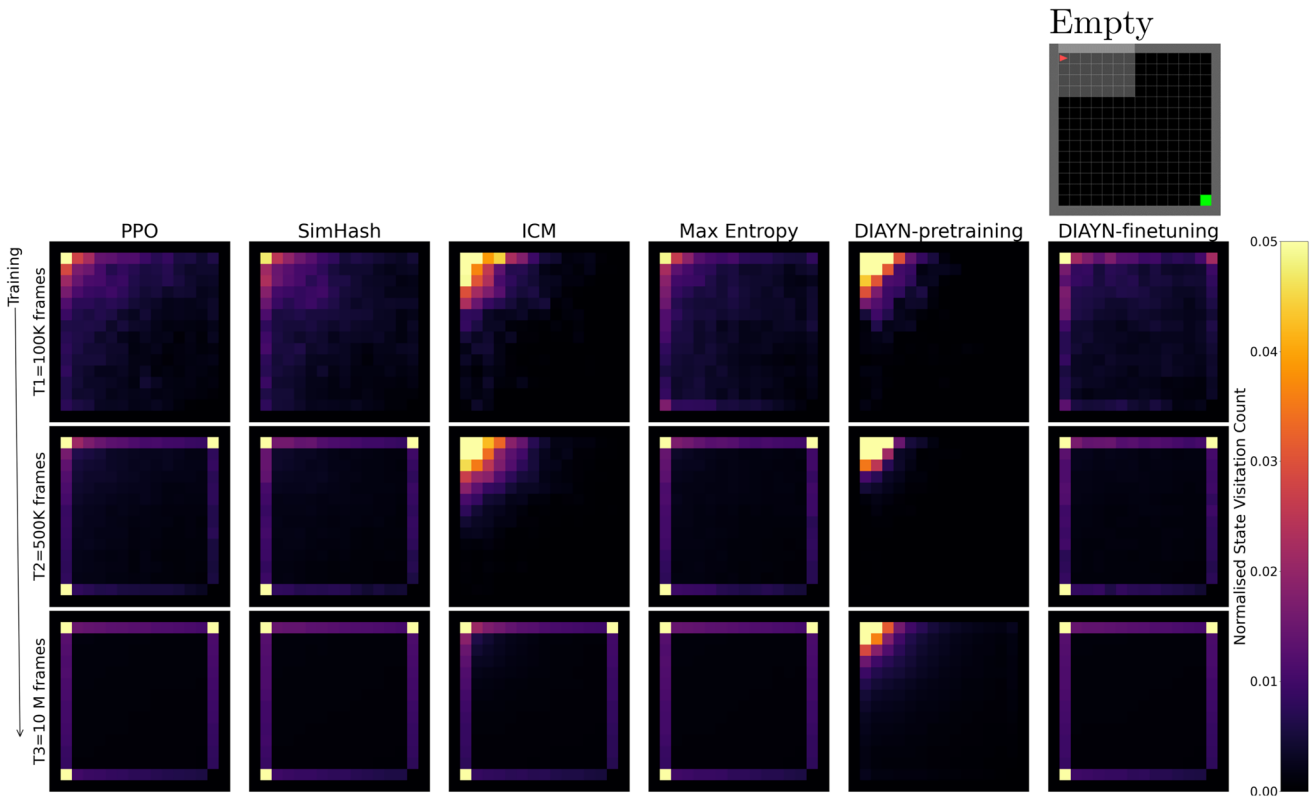


Fig. 13 State visitation count during training for 10 M frames on singleton Empty 16×16 environment with RGB observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

https://doi.org/10.1007/s00521-025-11340-0

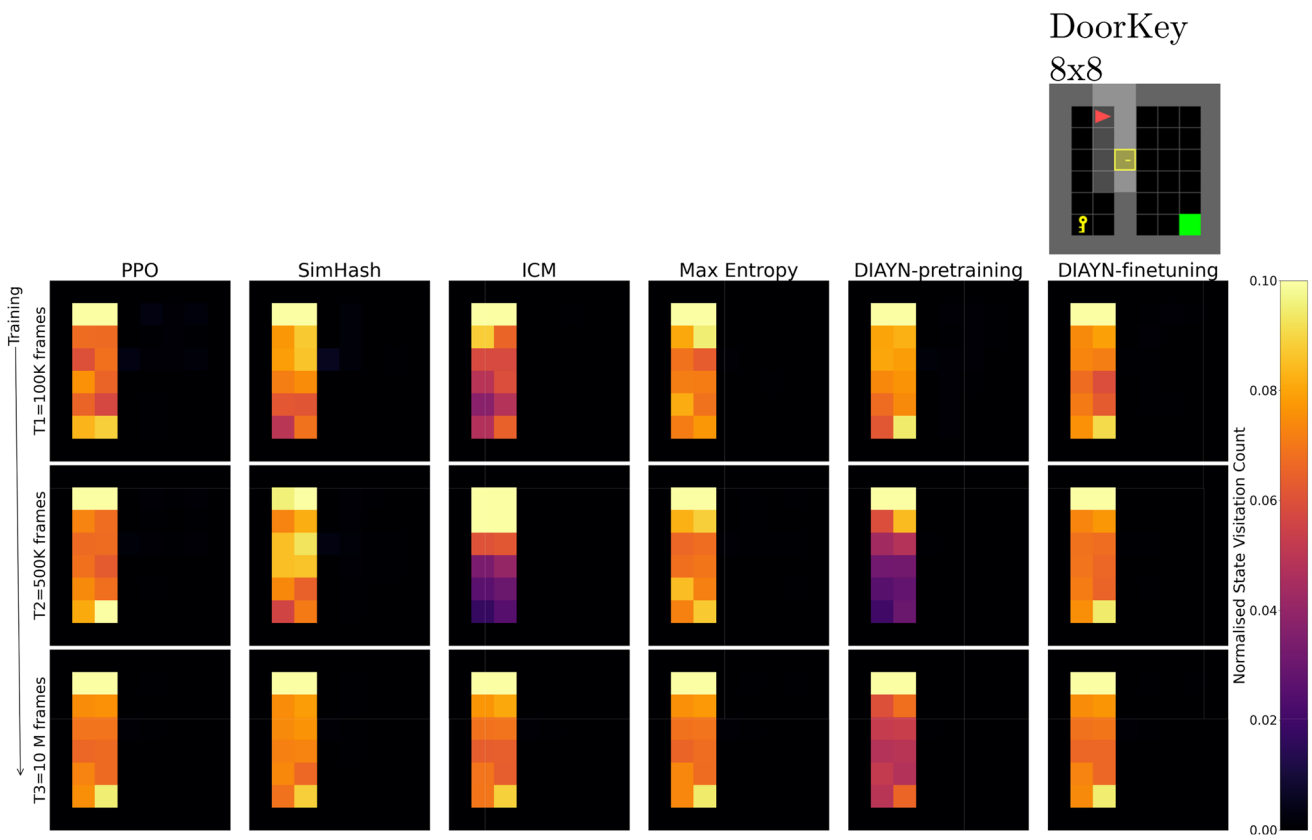


Fig. 14 State visitation count during training for 10 M frames on singleton DoorKey 8×8 environment with RGB observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames, and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

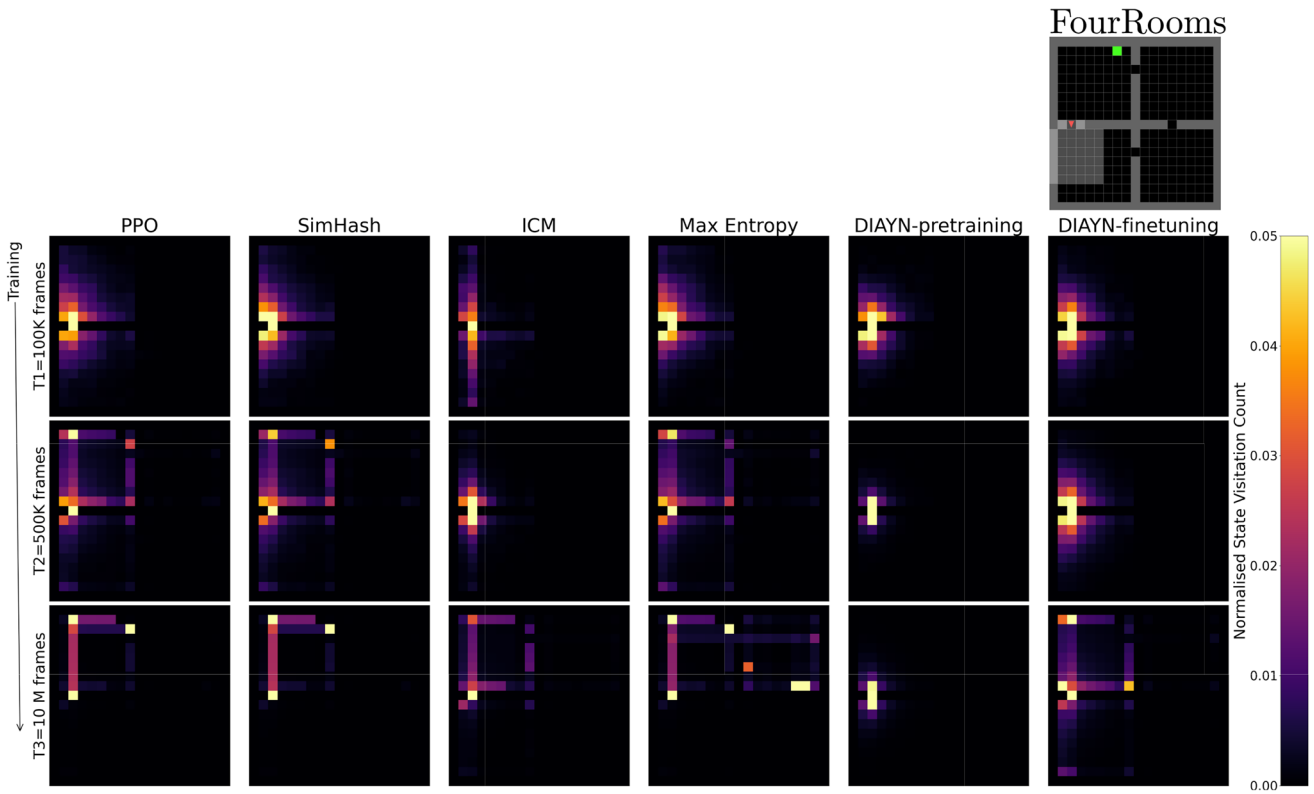


Fig. 15 State visitation count during training for 10 M frames on singleton FourRooms environment with RGB observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames, and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

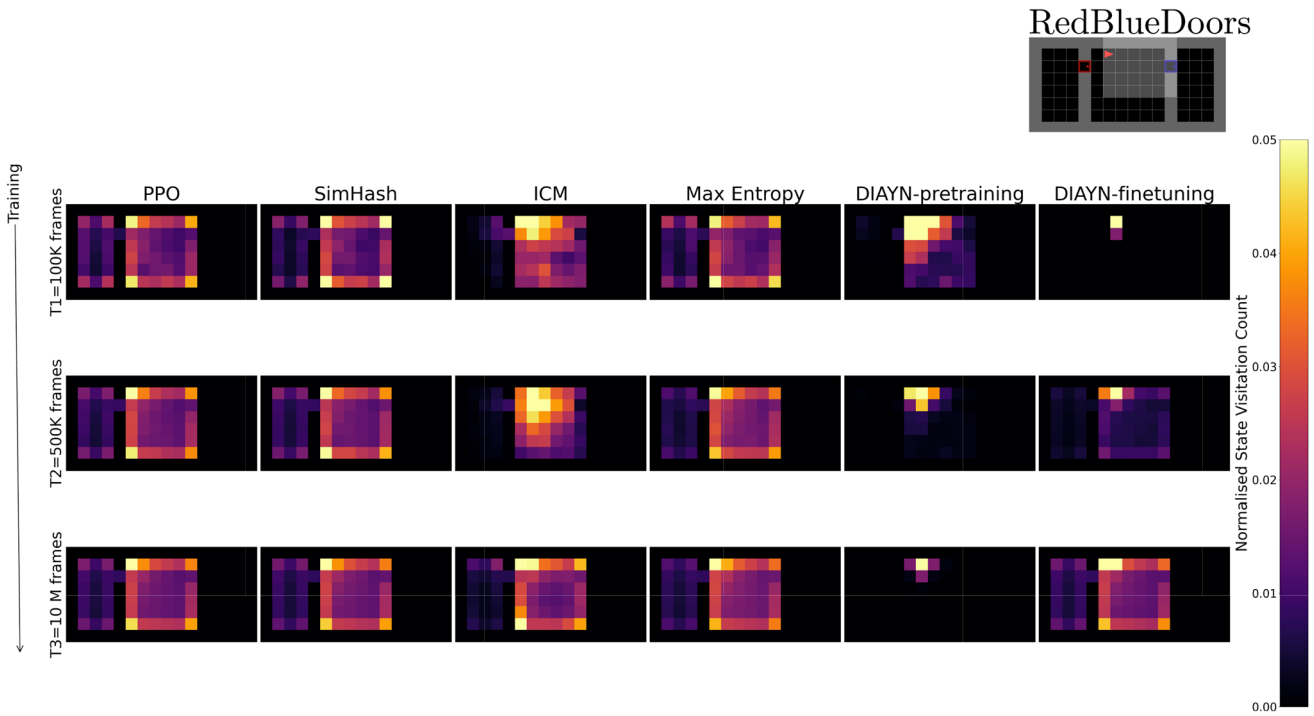


Fig. 16 State visitation count during training for 10 M frames on singleton RedBlueDoors environment with RGB observations. For each intrinsic reward method, snapshots of the heatmap are taken at three different timesteps T1: 100 K frames, T2: 500 K frames, and T3: 10 M frames. Color intensity represents the proportion of frames spent in each state, with high values capped for better visualization

Appendix E: DIAYN extrinsic

Initially, we evaluated DIAYN combined with extrinsic rewards, but it did not perform well because of the imbalance between discriminability and reward maximization (see Fig. 17). Recognizing that DIAYN is primarily intended for unsupervised pretraining of skills rather than simultaneous use with return maximization, we decided to split the training budget between pretraining and finetuning.

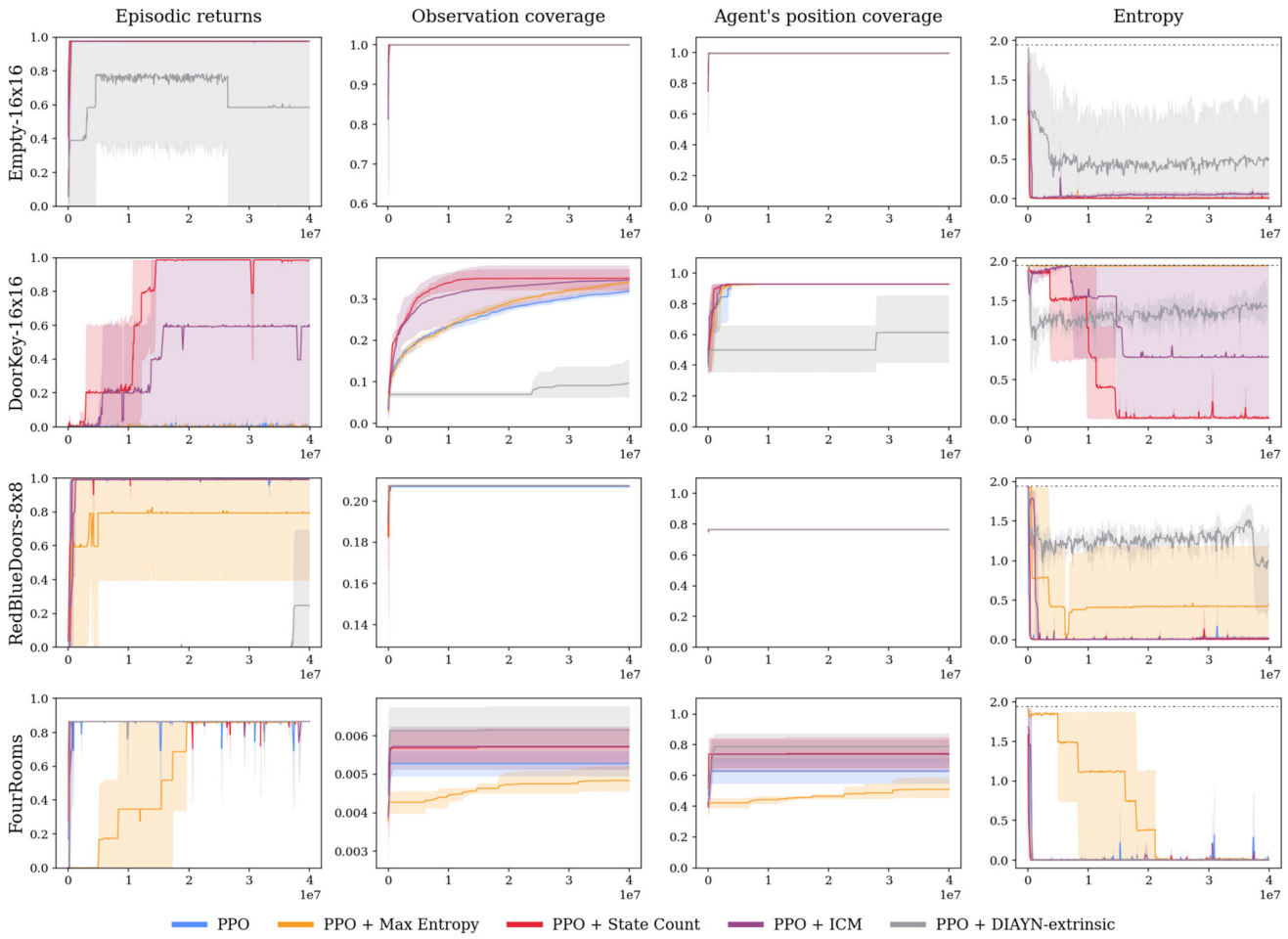


Fig. 17 Results for DIAYN combined with extrinsic rewards on grid encoding observation space

Acknowledgements The authors extend their gratitude to Alain Andrés Fernández for engaging in early discussions during the project and for sharing valuable materials. They also thank Corina Caraconcea for her assistance in visualizing heatmaps that provide intuition about the results.

Author contributions (1) Aya Kayal is the lead author who conceived, planned, carried out the experiments, and wrote the manuscript with inputs from other authors. (2) Eduardo Pignatelli contributed to developing conceptual ideas and designing experiments, provided technical advice, assisted in interpreting and plotting the results, and provided feedback on the manuscript. (3) Laura Toni is the project supervisor who provided critical feedback, and contributed to shaping the research, analysis, and manuscript. All authors reviewed the results and approved the final version of the manuscript.

Data availability This manuscript has no associated data.

Declarations

Conflict of interest The authors have no relevant financial or nonfinancial interests to disclose.

References

1. Ladosz P, Weng L, Kim M, Oh H (2022) Exploration in deep reinforcement learning: a survey. Inf Fusion 85:1–22
2. Gou SZ, Liu Y (2019) DQN with model-based exploration: efficient learning on environments with sparse rewards. arXiv preprint arXiv:1903.09295
3. Wiering MA (1999) Explorations in efficient reinforcement learning. PhD thesis, University of Amsterdam
4. François-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J et al (2018) An introduction to deep reinforcement learning. Found Trends Mach Learn 11(3–4):219–354
5. Amin S, Gomrokchi M, Satija H, Hoof H, Precup D (2021) A survey of exploration methods in reinforcement learning. arXiv preprint arXiv:2109.00157
6. Singh S, Lewis RL, Barto AG, Sorg J (2010) Intrinsically motivated reinforcement learning: an evolutionary perspective. IEEE Trans Auton Ment Dev 2(2):70–82
7. Chentanez N, Barto A, Singh S (2004) Intrinsically motivated reinforcement learning. Adv Neural Inf Process Syst 17
8. Aubret A, Matignon L, Hassas S (2019) A survey on intrinsic motivation in reinforcement learning. arXiv preprint arXiv:1908.06976
9. Ryan RM, Deci EL (2000) Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am Psychol 55(1):68
10. Oudeyer P-Y, Kaplan F (2007) What is intrinsic motivation? A typology of computational approaches. Front Neurorobot 1:6
11. Colas C, Karch T, Sigaud O, Oudeyer P-Y (2022) Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey. J Artif Intell Res 74:1159–1199
12. Siddique N, Dhakan P, Rano I, Merrick K (2017) A review of the relationship between novelty, intrinsic motivation and reinforcement learning. Paladyn, J Behav Robot 8(1):58–69
13. Bellemare M, Srinivasan S, Ostrovski G, Schaul T, Saxton D, Munos R (2016) Unifying count-based exploration and intrinsic motivation. Adv Neural Inf Process Syst 29
14. Tang H, Houthooft R, Foote D, Stooke A, Xi Chen O, Duan Y, Schulman J, DeTurck F, Abbeel P # exploration: a study of count-based exploration for deep reinforcement learning. Adv Neural Inf Process Syst 30 (2017)
15. Burda Y, Edwards H, Storkey A, Klimov O (2018) Exploration by random network distillation. arXiv preprint arXiv:1810.12894
16. Badia AP, Sprechmann P, Vitvitskyi A, Guo D, Piot B, Kapturowski S, Tieleman O, Arjovsky M, Pritzel A, Bolt A, et al.: Never give up: learning directed exploration strategies. arXiv preprint arXiv:2002.06038 (2020)
17. Pathak D, Agrawal P, Efros AA, Darrell T (2017) Curiosity-driven exploration by self-supervised prediction. In: International conference on machine learning, pp 2778–2787. PMLR
18. Houthooft R, Chen X, Duan Y, Schulman J, De Turck F, Abbeel P (2016) Vime: variational information maximizing exploration. Adv Neural Inf Process Syst 29
19. Aubret A, Matignon L, Hassas S (2023) An information-theoretic perspective on intrinsic motivation in reinforcement learning: a survey. Entropy 25(2):327
20. Eysenbach B, Gupta A, Ibarz J, Levine S (2018) Diversity is all you need: learning skills without a reward function. arXiv preprint arXiv:1802.06070
21. Cohen A, Yu L, Qiao X, Tong X (2019) Maximum entropy diverse exploration: Disentangling maximum entropy reinforcement learning. arXiv preprint arXiv:1911.00828
22. Gregor K, Rezende DJ, Wierstra D (2016) Variational intrinsic control. arXiv preprint arXiv:1611.07507
23. Bettini M, Kortvelesy R, Prorok A (2024) Controlling behavioral diversity in multi-agent reinforcement learning. arXiv preprint arXiv:2405.15054
24. Osa T, Harada T (2024) Discovering multiple solutions from a single task in offline reinforcement learning. arXiv preprint arXiv:2406.05993
25. Kumar S, Kumar A, Levine S, Finn C (2020) One solution is not all you need: few-shot extrapolation via structured MaxEnt RL. Adv Neural Inf Process Syst 33:8198–8210
26. Zahavy T, Schroecker Y, Behbahani F, Baumli K, Flennerhag S, Hou S, Singh S Discovering policies with DOMiNO: diversity optimization maintaining near optimality. arXiv preprint arXiv:2205.13521 (2022)
27. Grillotti L, Faldor M, León BG, Cully A (2024) Quality-diversity actor-critic: learning high-performing and diverse behaviors via value and successor features critics. arXiv preprint arXiv:2403.09930
28. McKee KR, Leibo JZ, Beattie C, Everett R (2022) Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. Auton Agent Multi-Agent Syst 36(1):21
29. Chen W, Huang S, Chiang Y, Pearce T, Tu W-W, Chen T, Zhu J (2024) DGPO: discovering multiple strategies with diversity-guided policy optimization. In: Proceedings of the AAAI conference on artificial intelligence, vol 38, pp 11390–11398
30. Cideron G, Agostinelli A, Ferret J, Girgin S, Elie R, Bachem O, Perrin S, Ramé A (2024) Diversity-rewarded CFG distillation. arXiv preprint arXiv:2410.06084

31. Hao J, Yang T, Tang H, Bai C, Liu J, Meng Z, Liu P, Wang Z (2023) Exploration in deep reinforcement learning: from single-agent to multiagent domain. IEEE Trans Neural Netw Learn Syst
32. Andres A, Villar-Rodriguez E, Del Ser J (2022) 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, pp 201–220. Springer
33. Strehl AL, Littman ML (2008) An analysis of model-based interval estimation for Markov decision processes. J Comput Syst Sci 74(8):1309–1331
34. Raileanu R, Rocktäschel T (2020) Ride: rewarding impact-driven exploration for procedurally-generated environments. arXiv preprint arXiv:2002.12292
35. Taiga AA, Fedus W, Machado MC, Courville A, Bellemare MG (2021) On bonus-based exploration methods in the arcade learning environment. arXiv preprint arXiv:2109.11052
36. Fortunato, M., Azar, M.G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al.: Noisy networks for exploration. arXiv preprint arXiv:1706.10295 (2017)
37. Bellemare MG, Naddaf Y, Veness J, Bowling M (2013) The arcade learning environment: an evaluation platform for general agents. J Artif Intell Res 47:253–279
38. Yuan M, Castanyer RC, Li B, Jin X, Berseth G, Zeng W (2024) Rlexplore: accelerating research in intrinsically-motivated reinforcement learning. arXiv preprint arXiv:2405.19548
39. Pathak D, Gandhi D, Gupta A (2019) Self-supervised exploration via disagreement. In: International conference on machine learning, pp 5062–5071. PMLR
40. Seo Y, Chen L, Shin J, Lee H, Abbeel P, Lee K (2021) State entropy maximization with random encoders for efficient exploration. In: International conference on machine learning, pp 9443–9454 . PMLR
41. Henaff M, Raileanu R, Jiang M, Rocktäschel T (2022) Exploration via elliptical episodic bonuses. Adv Neural Inf Process Syst 35:37631–37646
42. Laskin M, Yarats D, Liu H, Lee K, Zhan A, Lu K, Cang C, Pinto L, Abbeel P (2021) Urlb: Unsupervised reinforcement learning benchmark. arXiv preprint arXiv:2110.15191
43. Wang K, Zhou K, Kang B, Feng J, Shuicheng Y (2022) Revisiting intrinsic reward for exploration in procedurally generated environments. In: The eleventh international conference on learning representations
44. Henaff M, Jiang M, Raileanu R (2023) A study of global and episodic bonuses for exploration in contextual MDPs. arXiv preprint arXiv:2306.03236
45. Samvelyan M, Kirk R, Kurin V, Parker-Holder J, Jiang M, Hambro E, Petroni F, Kuttler H, Grefenstette E, Rocktäschel T (2021) Minihack the planet: a sandbox for open-ended reinforcement learning research. In: Thirty-fifth conference on neural information processing systems datasets and benchmarks track
46. Lin T, Jabri A (2024) Mimex: intrinsic rewards from masked input modeling. Adv Neural Inf Process Syst 36
47. Zahavy T, O'Donoghue B, Desjardins G, Singh S (2021) Reward is enough for convex MDPS. Adv Neural Inf Process Syst 34:25746–25759
48. Chevalier-Boisvert M, Dai B, Towers M, Lazcano R, Willems L, Lahlou S, Pal S, Castro PS, Terry J (2023) Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. CoRR **abs/2306.13831**
49. Haarnoja T, Zhou A, Abbeel P, Levine S (2018) Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning, pp 1861–1870. PMLR
50. Liu J, Gu X, Liu S (2019) Policy optimization reinforcement learning with entropy regularization. arXiv preprint arXiv:1912.01557
51. McCallum AK (1996) Reinforcement learning with selective perception and hidden state. PhD thesis
52. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347
53. Ahmed Z, Le Roux N, Norouzi M, Schuurmans D (2019) Understanding the impact of entropy on policy optimization. In: International conference on machine learning, pp 151–160. PMLR
54. Liu Z, Li X, Kang B, Darrell T (2019) Regularization matters in policy optimization. arXiv preprint arXiv:1910.09191
55. Han S, Sung Y (2021) A max-min entropy framework for reinforcement learning. Adv Neural Inf Process Syst 34:25732–25745
56. Yang R, Bai C, Guo H, Li S, Zhao B, Wang Z, Liu P, Li X (2023) Behavior contrastive learning for unsupervised skill discovery. In: International conference on machine learning, pp 39183–39204. PMLR
57. Tolguenec P-AL, Besse Y, Teichteil-Konigsbuch F, Wilson DG, Rachelson E (2024)Exploration by learning diverse skills through successor state measures. arXiv preprint arXiv:2406.10127
58. Kim H, Lee BK, Lee H, Hwang D, Park S, Min K, Choo J (2024) Learning to discover skills through guidance. Adv Neural Inf Process Syst 36
59. Pignatelli E, Ferret J, Geist M, Mesnard T, Hasselt H, Toni L (2024) A survey of temporal credit assignment in deep reinforcement learning. Trans Mach Learn Res. Survey Certification
60. Todorov E, Erez T, Tassa Y (2012) Mujoco: a physics engine for model-based control. In: IEEE/RSJ international conference on intelligent robots and systems, pp 5026–5033. IEEE

61. Jiang M, Dennis M, Grefenstette E, Rocktäschel T (2023) Minimax: Efficient baselines for autocurricula in jax. arXiv preprint arXiv:2311.12716
62. Lin Z, D'Oro P, Nikishin E, Courville A (2024) The curse of diversity in ensemble-based exploration. arXiv preprint arXiv:2405.04342
63. Pong VH, Dalal M, Lin S, Nair A, Bahl S, Levine S (2019) Skew-fit: State-covering self-supervised reinforcement learning. arXiv preprint arXiv:1903.03698
64. Colas C, Fournier P, Chetouani M, Sigaud O, Oudeyer P-Y (2019) Curious: intrinsically motivated modular multi-goal reinforcement learning. In: International conference on machine learning, pp 1331–1340. PMLR
65. Lin Y, Lin F, Yang L, Zeng W, Liu Y, Wu P (2022) Context-aware reinforcement learning for course recommendation. Appl Soft Comput 125:109189
66. Kiani F, Saraç ÖF (2023) A novel intelligent traffic recovery model for emergency vehicles based on context-aware reinforcement learning. Inf Sci 619:288–309
67. Thakoor N, Bhanu B (2013) Context-aware reinforcement learning for re-identification in a video network. In: Seventh international conference on distributed smart cameras (ICDSC), pp 1–6 . IEEE
68. Jin C, Allen-Zhu Z, Bubeck S, Jordan MI (2018) Is Q-learning provably efficient? Adv Neural Inf Process Syst 31
69. Kolter JZ, Ng AY (2009) Near-Bayesian exploration in polynomial time. In: International conference on machine learning, pp 513–520
70. Zhang T, Xu H, Wang X, Wu Y, Keutzer K, Gonzalez JE, Tian Y (2021) Noveld: a simple yet effective exploration criterion. Adv Neural Inf Process Syst 34:25217–25230
71. Hazan E, Kakade S, Singh K, Van Soest A (2019) Provably efficient maximum entropy exploration. In: International conference on machine learning, pp 2681–2691. PMLR
72. Lee L, Eysenbach B, Parisotto E, Xing E, Levine S, Salakhutdinov R (2019) Efficient exploration via state marginal matching. arXiv preprint arXiv:1906.05274
73. Mutti M, Pratissoli L, Restelli M (2020) A policy gradient method for task-agnostic exploration. In: 4th lifelong machine learning workshop at ICML
74. Liu H, Abbeel P (2021) Behavior from the void: unsupervised active pre-training. Adv Neural Inf Process Syst 34:18459–18473
75. Kim D, Shin J, Abbeel P, Seo Y (2024) Accelerating reinforcement learning with value-conditional state entropy exploration. Adv Neural Inf Process Syst 36
76. Burda Y, Edwards H, Pathak D, Storkey A, Darrell T, Efros AA (2018) Large-scale study of curiosity-driven learning. arXiv preprint arXiv:1808.04355
77. Badia AP, Piot B, Kapturowski S, Sprechmann P, Vitvitskyi A, Guo ZD, Blundell C (2020) Agent57: outperforming the Atari human benchmark. In: International conference on machine learning, pp 507–517. PMLR
78. Savinov N, Raichuk A, Marinier R, Vincent D, Pollefeys M, Lillicrap T, Gelly S (2018) Episodic curiosity through reachability. arXiv preprint arXiv:1810.02274
79. Wang Y, Yang M, Dong R, Sun B, Liu F, et al (2024) Efficient potential-based exploration in reinforcement learning using inverse dynamic bisimulation metric. Adv Neural Inf Process Syst 36
80. Zhu A, Zhang P-F, Qiu R, Zheng Z, Huang Z, Shao J (2024) Abstract and explore: a novel behavioral metric with cyclic dynamics in reinforcement learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 38, pp 17150–17158
81. Machado MC, Bellemare MG, Bowling M (2020) Count-based exploration with the successor representation. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 5125–5133
82. Yu C, Burgess N, Sahani M, Gershman SJ (2024) Successor-predecessor intrinsic exploration. Adv Neural Inf Process Syst 36
83. Eysenbach B, Levine S (2021) Maximum entropy RL (provably) solves some robust RL problems. arXiv preprint arXiv:2103.06257
84. Hong Z-W, Shann T-Y, Su S-Y, Chang Y-H, Fu T-J, Lee C-Y (2018) Diversity-driven exploration strategy for deep reinforcement learning. Adv Neural Inf Process Syst 31
85. Flet-Berliac Y, Ferret J, Pietquin O, Preux P, Geist M (2021) Adversarially guided actor-critic. arXiv preprint arXiv:2102.04376
86. Conti E, Madhavan V, Petroski Such F, Lehman J, Stanley K, Clune J (2018) Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. Adv Neural Inf Process Syst 31
87. Parker-Holder J, Pacchiano A, Choromanski KM, Roberts SJ (2020) Effective diversity in population based reinforcement learning. Adv Neural Inf Process Syst 33:18050–18062
88. Li P, Hao J, Tang H, Fu X, Zhen Y, Tang K (2024) Bridging evolutionary algorithms and reinforcement learning: a comprehensive survey on hybrid algorithms. IEEE Trans Evol Comput
89. Gaya J-B, Soulier L, Denoyer L (2021) Learning a subspace of policies for online adaptation in reinforcement learning. arXiv preprint arXiv:2110.05169

Authors and Affiliations

Aya Kayal[1] · Eduardo Pignatelli[1] · Laura Toni[1]

✉ Aya Kayal
aya.kayal.21@ucl.ac.uk

Eduardo Pignatelli
e.pignatelli@ucl.ac.uk

Laura Toni
l.toni@ucl.ac.uk

[1] Electronic and Electrical Engineering Department, University College London, London, UK