Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2025)

Minding Motivation: The Effect of Intrinsic Motivation on Agent Behaviors

Leonardo Villalobos-Arias, Grant Forbes, Jianxun Wang, David L Roberts, Arnav Jhala

North Carolina State University
Department of Computer Science
Raleigh, North Carolina, USA
{lvillal, gforbes, jwang75, dlrober4, ahjhala}@ncsu.edu

Abstract

Games are challenging for Reinforcement Learning (RL) agents due to their reward sparsity, as rewards are only obtainable after long sequences of deliberate actions. Intrinsic Motivation (IM) methods—which introduce exploration rewards—are an effective solution to reward sparsity. However, IM also causes an issue known as 'reward hacking', where the agent optimizes for the new reward at the expense of properly playing the game. The larger problem is that reward hacking itself is largely unknown; there is no answer to whether, and to what extent, IM rewards change the behavior of RL agents. This study takes a first step by empirically evaluating the impact on behavior of three IM techniques on the MiniGrid game-like environment. We compare these IM models with Generalized Reward Matching (GRM), a method that can be used with any intrinsic reward function to guarantee optimality. Our results suggest that IM causes noticeable change by increasing the initial rewards, but also altering the way the agent plays, and that GRM mitigated reward hacking in some scenarios.

Introduction

The phrase "limitation breeds creativity" has never been more applicable to artificial intelligence in video games than now. Some of the most important breakthroughs in Reinforcement Learning (RL) research have been reached through game-playing agents (Mnih et al. 2013, 2015; Schulman et al. 2017; Vinyals et al. 2019). Reward-sparse games are characteristically difficult for RL to learn due to the long sequence of actions required to both discover and then properly attribute sparse rewards (Pathak et al. 2017; Huang and Ontañón 2020). Traditional ϵ-greedy exploration will fail to find the goal state, and thus the sparse reward, as it will likely never stumble into the goal state and fail to construct a policy (Pathak et al. 2017). In contrast, an agent motivated by intrinsic rewards will thoroughly explore the environment and eventually reach the real, extrinsic reward. Research on Intrinsic Motivation (IM)—a method that enhances the environment with exploration rewards—has led to RL agents that can perform well on these hard games (Burda et al. 2018).

Intrinsic Motivation, while useful and even necessary, has its own set of drawbacks. IM is prone to 'reward hacking,' a phenomenon where the agent optimizes for the shaped reward at the expense of the actual reward (Forbes et al. 2024a). Previous benchmark studies have demonstrated that IM agents underperform against ϵ-greedy exploration, since the IM agent is optimizing for both intrinsic and extrinsic (real) rewards and will likely not find a policy that is optimal for the real reward alone (Taïga et al. 2021). As an example of reward hacking, Burda et al. (2019) describes their IM agents positioning themselves next to hazards, a behavior aptly named "dancing with skulls," for the intrinsic reward associated with their inherent rarity at the expense of playing the game. A related issue is the 'noisy-TV' problem (Burda et al. 2018), where the agent, distracted by intrinsic rewards, disregards the search for the real reward. Ongoing research on optimality-preserving methods harnesses the benefits of IM while reducing its negative effects (Raileanu and Rocktäschel 2019; Behboudian et al. 2022; Forbes et al. 2024a).

A larger problem is that the effects of Intrinsic Motivation on the way RL agents behave are largely unknown. The state-of-the-art experiment designs (Taiga et al. 2019; Andres, Villar-Rodriguez, and Del Ser 2022; Forbes et al. 2024b) focus only on the maximization of rewards over time. While this is a good indicator for performance, it is a poor evaluation metric for exploration (Ladosz et al. 2022). For example, an IM method could result in a lower final average reward, yet it could be 'closer' to finding the optimal policy than an agent trained with no IM. For the more complex game environments, said optimal policy is completely unknown even to expert human players. A common practice in IM RL literature is to benchmark methods on environments where RL can not properly learn without IM, meaning the optimal policy is impossible to know in that case. With no baseline behavior, evaluating the policy-invariance of IM becomes impossible. For these reasons, there is no answer to what extent, and under which conditions, intrinsic rewards affect the final behavior of game-playing agents.

A related problem is the over-reliance on analysis of rewards over qualitative analysis, such as policy visualization. While it is the intuitive approach (higher reward equals better model), it is a poor tool for causality and policy optimality. For instance, the "dancing with skulls" behavior was

discovered by visualizing the agent's policy (Burda et al. 2019), and yet we are only aware of a handful of papers that actively check the effects of IM using such methods (Huang and Ontañón 2020; Le et al. 2024; Raileanu and Rocktäschel 2019; Kayal, Pignatelli, and Toni 2025). Behavior analysis of RL models can reveal more about why certain methods work (or not), what advantages and disadvantages they provide, which corner cases change their performance, among other benefits. Hypothetically, it could be possible that 'reward hacking' is not necessarily an issue due to the eventual exhaustion of intrinsic rewards. Similarly, IM methods might hypothetically require more training to converge to optimality over traditional RL. These and similar questions remain open due to the lack of behavior-focused research.

To address these gaps in the literature, this paper proposes an empirical evaluation of the effect of intrinsic motivation techniques on the behavior of reinforcement learning agents. We measured the impact of these methods in terms of both reward performance and exhibited behavior. The protocol of this study was based on Kayal, Pignatelli, and Toni (2025), from which we selected three traditional IM methods to evaluate: State Counting, Max Entropy, and Intrinsic Curiosity Model (ICM). As a representative of policy-invariant methods, we selected Generalized Reward Matching (GRM, specifically D-GRM) (Forbes et al. 2024b) and combined it with these three IM sources. We trained the agents on Minigrid (Chevalier-Boisvert et al. 2023), a simplification of game-like environments, since it allows behavioral analysis in the form of policy visualization. An important distinction with previous IM evaluations is that our selection of environments can be learned with no intrinsic rewards. To talk about behavior in RL and how different IM methods result in different behaviors, it is necessary to have a baseline policy. Our main contribution is an empirical analysis of the policy variance of various IM techniques. The results of this experiment also double as a benchmark of GRM, which previously was tested only on Montezuma's Revenge (Forbes et al. 2024a). We expanded on the work by Kayal, Pignatelli, and Toni (2025) with our change of analytical focus to agent behavior and optimality-preservation, the addition of GRM to the evaluation, and adjustment of the methods and environment to suit the baseline model to properly learn.

Related Work

Intrinsic Motivation is a subfield of reward shaping that augments a sparse-reward environment with an additional reward function based on "intrinsic" goals, which are often complex, non-Markovian shaping rewards meant to generalize well across environments, rather than being tailored to a particular environment. These rewards are often based on psychological concepts (Oudeyer and Kaplan 2007) such as curiosity, empowerment, novelty, or skill-learning (Burda et al. 2018; Mohamed and Jimenez Rezende 2015; Colas et al. 2022).

Count-based rewards award the agent with an intrinsic reward proportionally inverse to the number of times a state has been visited (Strehl and Littman 2008). Andres, Villar-Rodriguez, and Del Ser (2022) present a simple count-based reward $r_i = 1/\sqrt{(N_s)}$, where N_s is the number of times state s has been visited so far. These types of methods are most effective in environments with small, discrete state spaces. A similar type of reward based on state novelty but scaled to high-dimensional environments is Random Network Distillation (RND) (Burda et al. 2019). RND grants a reward equal to the loss of a neural network model trained to predict a randomly initialized state encoder, resulting in behavior similar to counting but grouping visually similar states through a convolutional neural network. Other examples of novelty-based techniques include Discriminative-model-based Episodic Intrinsic Reward (DEIR) (Wan et al. 2023), Never Give Up (Badia et al. 2019), and NovelD (Zhang et al. 2021).

Curiosity-based methods reward the agent for unexpected results, in contrast to unexpected states. A simple technique is max entropy (Liu, Gu, and Liu 2020; Kayal, Pignatelli, and Toni 2025), where the agent is granted a reward equal to its policy entropy in order to incentivize stochastic policies (i.e., exploration over exploitation). Intrinsic Curiosity Model (ICM) is such a technique (Pathak et al. 2017). Based on a learned state-embedding and an inverse-dynamics model, ICM trains a forward-dynamics model on the next state based on the agent's action, granting an intrinsic reward according to its loss.

There is ongoing research to mitigate the side effects of intrinsic motivation. Huang and Ontañón (2020) propose an algorithm called Action Guidance, which consists of learning separate policies for the real and shaped (intrinsic) rewards, tested on the MicroRTS environment (Ontañón et al. 2018), though it requires the use of off-policy gradient methods. However, they do not compare Action Guidance against any existing IM methods (only a hand-crafted shaped-reward function). The EIPO algorithm proposed in Chen et al. (2022) automatically scales the intrinsic reward coefficient by augmenting it when exploration is necessary, and vice-versa. Although their benchmark is solid, they lack any type of policy analysis. Le et al. (2024) introduce the concept of "surprise novelty" to mitigate the noisy TV problem. The authors visualized the policy and intrinsic rewards of the agents, but did not compare the policies of different agents. Raileanu and Rocktäschel (2019) introduces Rewarding Impact-Driven Exploration (RIDE) as a type of IM method designed for one-shot (a state is seldom visited twice) observation problems, such as procedurally generated environments. Similar to the previous study, they plot the intrinsic reward functions they compared, but they go a step further and compare the policies. Behboudian et al. (2022) create, based on the concept of potential-based rewards, the Policy-Invariant Explicit Shaping (PIES) algorithm that similarly diminishes the intrinsic reward to guarantee policy invariance by the end of training. As with most IM literature, their analysis focuses only on episodic returns. Following a similar line of research, Forbes et al. (2024a,b, 2025) extend the body of research of intrinsic motivation with three optimality-preserving algorithms: Potential-Based Intrinsic Motivation (PBIM), Generalized Reward Matching (GRM), and Action-Dependent Optimality-Preserving Shaping (ADOPS). While they benchmark their techniques

against other optimality-preserving methods, they only use RND rewards and no behavioral analysis.

Aside from the empirical analysis that often accompanies the proposal of a novel method, there is a limited number (Ladosz et al. 2022) of studies within RL that benchmark existing IM methods, and particularly existing optimality-preserving IM methods. Taiga et al. (2019) perform a study on the Atari Learning Environment, where they rank state count, ICM, RND, and NoisyNets. Laskin et al. (2021) create a benchmark suite for unsupervised RL, including IM methods. Andres, Villar-Rodriguez, and Del Ser (2022) perform an empirical evaluation of intrinsic motivation hyperparameters in the MiniGrid environment. Lastly, Kayal, Pignatelli, and Toni (2025) evaluates four intrinsic motivation algorithms on the MiniGrid environment by assessing how they impact the behavior of the RL agent. None of these prior empirical evaluations, however, has focused on empirically benchmarking and comparing prior IM methods on how their ability to preserve optimality. In this paper, we present such an evaluation.

Experimental Design

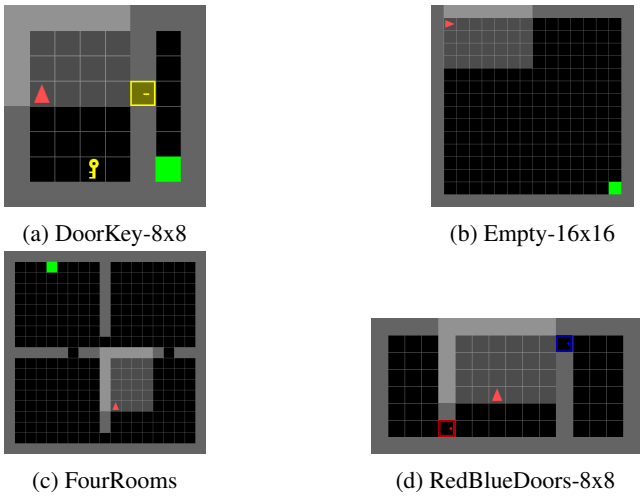
The main objective of this evaluation is to empirically evaluate the behavior of reinforcement agents when trained using different intrinsic motivation techniques. We measure this change of behavior on two dimensions: 1) return (real/extrinsic reward) performance, and 2) exhibited policy behavior.

We base our experimental design on a recent study (Kayal, Pignatelli, and Toni 2025), where the authors analyze how IM impacts the early exploration of RL agents. We shift focus away from emphasis on exploration and on the different levels of diversity of IM, and instead emphasize regular (policy-altering) and policy-invariant IM and their impact on behavior. Specifically, we analyze whether IM improves exploration and/or alters the final policy found by the agent. Our implementation is a modified version of the publicly available DEIR method (Wan et al. 2023), and is publicly available on: https://github.com/lyonva/bad-apple/releases/tag/mindingmotivation. Hereon, we refer to (Kayal, Pignatelli, and Toni 2025) as the protocol study.

Environment

We use four MiniGrid (Chevalier-Boisvert et al. 2023) reward-sparse environments, where the agent only gets a non-zero reward by the end of a successful episode. To incentivize efficiency, the environment (extrinsic) reward is inversely proportional to the number of time steps that the agent requires to get to the goal. We train agents in the following environments, which are shown in Figure 1:

- **DoorKey-8x8** requires getting to the goal tile, which is in a separate room behind a locked door. The agent thus must learn to pick up a key to open the door to proceed to the goal.
- **Empty-16x16** has the objective of getting to the goal tile, which is always on the bottom right part of the map. The entirety of the map is empty, except for the outermost wall tiles.



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

(c) FourRooms (d) RedBlueDoors-8x8

Figure 1: MiniGrid maps used in this experiment.

- **FourRooms** features four connected rooms, where both the goal tile and the agent's initial position are randomized. Reaching the goal often requires the agent to visit and explore two or more rooms.
- **RedBlueDoors-8x8** features a central room with a red door on the left side and a blue door on the right side. The agent must open the doors in this order.

These reward-sparse environments were chosen as they are simple enough that an agent will not require IM to learn an effective policy, but still can benefit from IM to learn faster.

The MiniGrid environment is partially observable; the agent has access to a $(7 \times 7 \times 3)$ observation of the tiles in front of it and is not explicitly told its position (coordinates) within the grid. The actions available to an agent in MiniGrid are: turn left, turn right, move forward, pickup, drop, and toggle.

Model Architecture

Following the protocol study, we used Proximal Policy Optimization (Schulman et al. 2017) as the base learning algorithm due to its broad applications in the RL literature, availability of implementation, and robustness. The model has a shared CNN to process MiniGrid observations, which is comprised of three layers: 16 filters, 32 filters, and 64 filters—all of which are sized 2×2 and use ReLU activation. The output of this network is then passed to the actor network and two critic networks, for independent estimation of intrinsic and extrinsic motivation values. The three networks have a single hidden layer of 64 units with ReLU activation. The actor outputs the probability of each of the seven actions, and the two value networks output the prediction of the extrinsic/intrinsic rewards.

We trained seven variants of this model: PPO with no IM (baseline), three variants of PPO with IM, and three variants of PPO with IM and GRM. We trained each model on each environment for a total of 20.48 million frames (1,000 rollouts), and repeated this process for a total of ten runs

# parallel environments	16
# frames per rollout	128
# epochs	4
Batch size	256
Discount γ	0.99
Learning rate	1×10^{-4}
Entropy regularization	5×10^{-4}
Value loss coefficient	0.5
PPO clipping factor	0.2
Gradient clipping	0.5

Table 1: PPO hyperparameters.

	DK	Empty	FR	RBD
State Count	1	0.01	1	1
Max Entropy	0.001	5×10^{-5}	1×10^{-5}	1×10^{-6}
ICM	0.1	0.01	0.05	1×10^{-6}
GRM+SC	1	0.01	0.05	0.1
GRM+ME	0.001	5×10^{-5}	1×10^{-5}	1×10^{-6}
GRM+ICM	0.1	0.01	0.05	1×10^{-6}

Table 2: Chosen values for intrinsic reward coefficient β. The value is set to 0 for the no-IM model.

per combination of map and IM method. We evaluated three intrinsic motivation methods:

- **State Count** (SC) grants a reward $1/\sqrt{(N_s)}$, where N_s is the number of times state s has been visited so far.
- **Max Entropy** (ME) awards the agent with an intrinsic reward equal to the policy network's entropy.
- **ICM** with one hidden layer of 256 units with ReLU activation, a learning rate of 3×10^{-4}, and a state encoder with the same CNN architecture as the PPO model.

Table 1 shows the selected values for the main PPO hyperparameters. In addition, we manually tuned, per map and method, the hyperparameter value of the intrinsic reward coefficient β in the range $[1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001, ..., 5 \times 10^{-6}, 1 \times 10^{-6}]$. Table 2 shows the final values for β.

Evaluation

We evaluate with performance metrics and behavioral analysis with visualizations of in-environment behavior.

We capture—per training rollout, and aggregated over all parallel actors—two evaluation metrics during training:

- **Episodic return:** The average reward obtained per elapsed episode averaged across all actors. It acts as a measure of how effective the policy is. Higher is better.
- **Position coverage:** The percentage of unique grid positions—(x, y) coordinates—visited across all actors. The total count of visited positions is divided across the number of tiles that are possible to visit in each map. It acts as a measure of how much an agent is exploring. Higher is better.

	DK	Empty	FR	RBD
State Count	0.40	0.42	**0.62**	**0.50**
Max Entropy	0.43	0.77	0.84	0.99
ICM	**0.31**	**0.27**	0.76	0.77
GRM+SC	0.37	0.79	0.72	0.61
GRM+ME	0.48	0.78	0.67	0.55
GRM+ICM	0.49	0.32	0.74	0.63

Table 3: Average policy divergence for IM models. Lowest values per map are highlighted.

For behavioral analysis, we recorded the state of the agent's model (network weights) at different points of training: 5% and 100%. To extract an approximate policy of each agent, we manually picked a map instance, shown in Figure 1, and simulated 5,000 steps per trained model while recording the agent's position. We then used heatmaps to visualize the frequency the agent stays at each position.

To determine whether IM causes changes in the final policy of an agent, and the effect size of that change, we propose the *policy divergence* metric. We randomly selected 10 map instances and simulated each of the fully trained models for 5,000 steps, recording their positions on the grid. We calculated policy divergence for each IM method as $\frac{1}{N}\sum_{i,j}|S_{(i,j)} - S'_{(i,j)}|$, where N is the total number of steps in the simulation (5,000), and S and S' are the visitation frequency of the grid position (i, j), for the IM method and the no-IM baseline respectively. Note that $\sum_{i,j} S_{(i,j)} = \sum_{i,j} S'_{(i,j)} = N$. We then averaged this metric over the 10 simulations. A policy divergence of 0 indicates the two compared policies are equivalent, whereas 2 indicates maximally divergent policies[1].

Results

Figures 2 and 3 respectively show the average episodic rewards and position coverage obtained by every trained agent, grouped by type of IM. Table 3 shows the average policy divergence for all IM methods. For readability, we will only show relevant samples of the heatmaps. The complete heatmap figures plus the data used to generate them are available in the following link: https://tinyurl.com/aiide25im.

IM Alters Final Policies

The use of intrinsic motivation methods (without GRM) often resulted in a distinct final policy. This is most noticeable in terms of return performance (Figure 2, first row). Due to the benefits of IM, it is normal for models trained with it to reach near-optimal rewards first. However, in the DoorKey and RedBlueDoors map, the final policy found by the baseline agent achieves noticeably different average return performance. The trend on these maps was that State Counting

[1]A current limitation is that, if an environment has more than one optimal policy, the policy divergence of a method may be higher than it should.

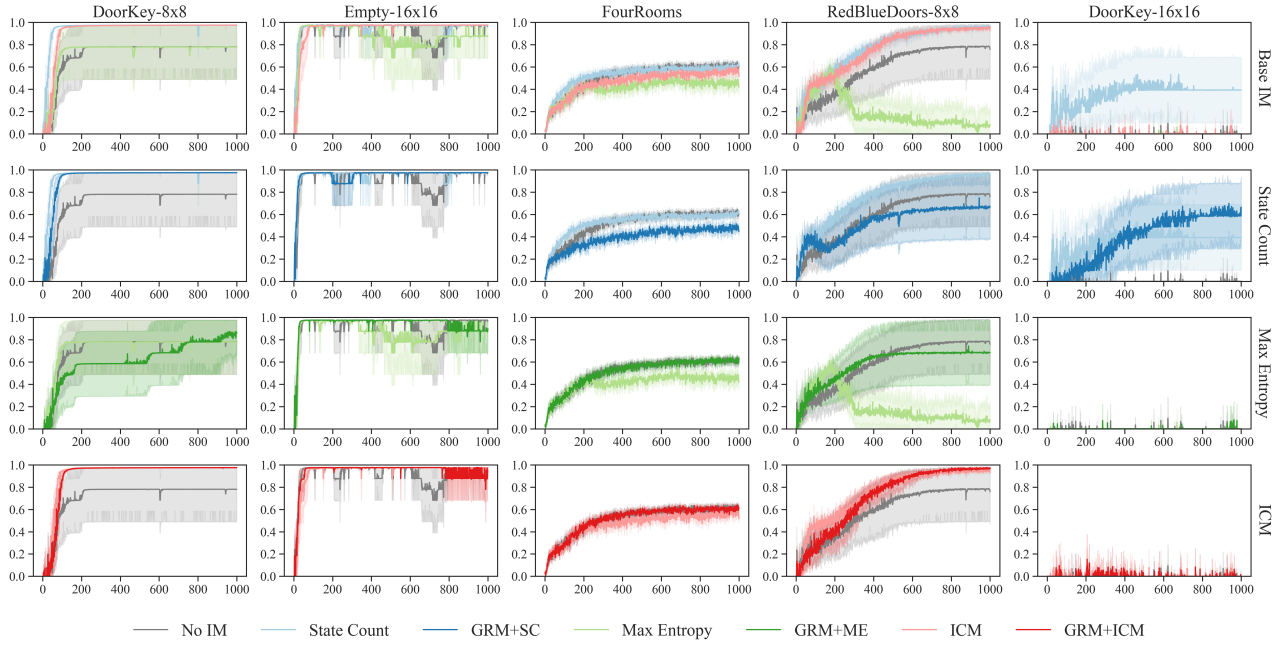


Figure 2: Episodic rewards per iteration of all the trained models. Columns group results by map and rows by type of IM: 1) non-GRM, 2) State Count, 3) Max Entropy, and 4) ICM. Rows two and onward include models with and without GRM. Results are averaged over 10 runs, and shading is the standard deviation.

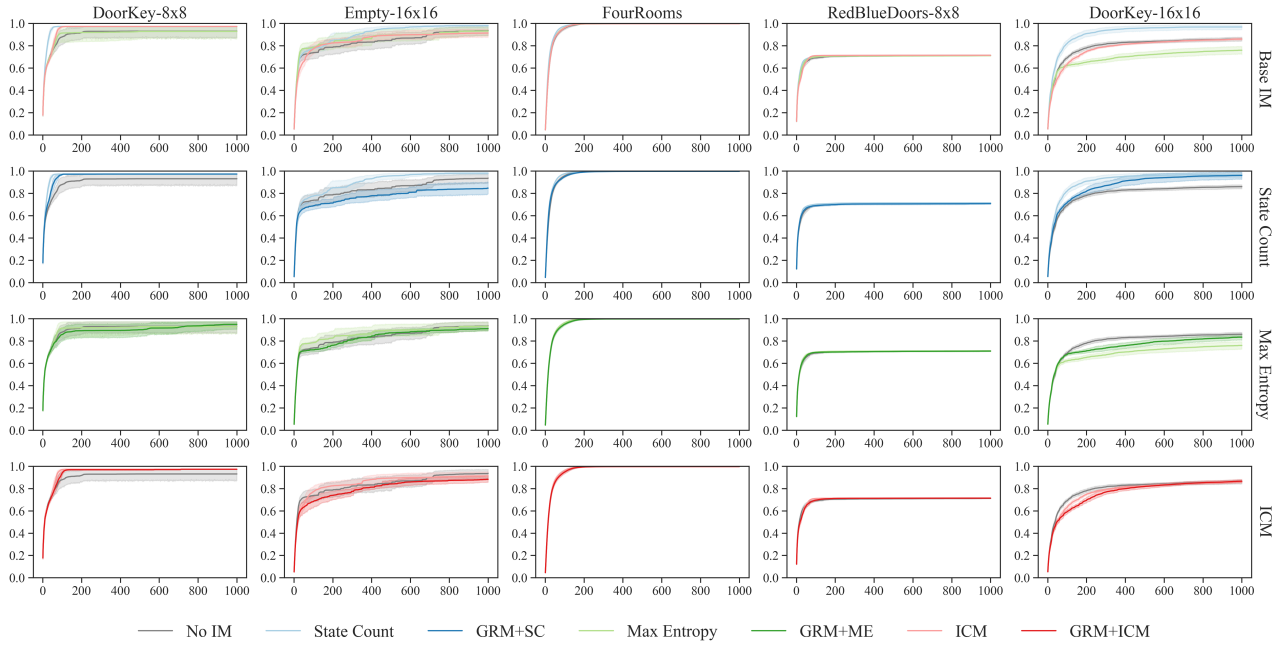
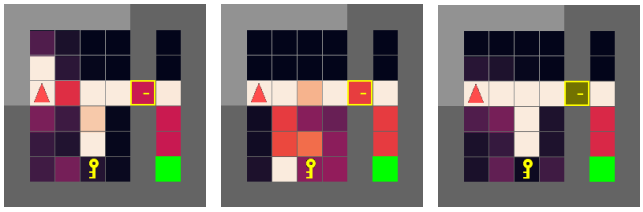
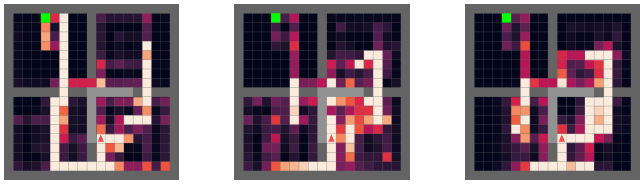


Figure 3: Position (tiles in grid) coverage per iteration of all the trained models. Columns group results by map and rows by type of IM: 1) non-GRM, 2) State Count, 3) Max Entropy, and 4) ICM. Rows two and onward include models with and without GRM. Results are averaged over 10 runs, and shading is the standard deviation.



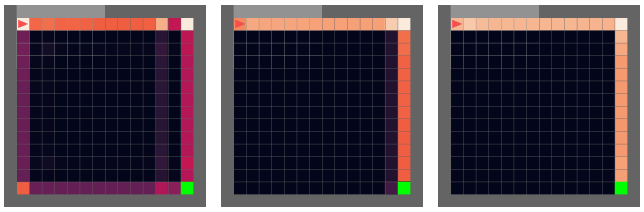
(a) No IM. (b) State Count. (c) GRM + SC.

Figure 4: DoorKey final agent behavior example. State Count policy results in a different average policy, whereas GRM reinforces the optimal path. A brighter color in a tile indicates higher visitation frequency.



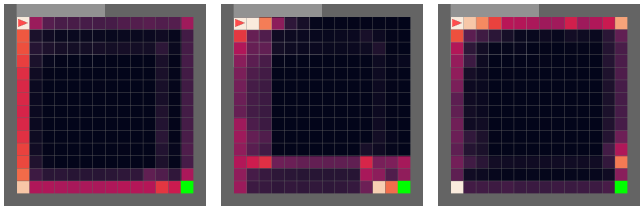
(a) No IM. (b) Max Entropy. (c) ICM.

Figure 5: FourRooms final agent behavior example. Max Entropy and ICM result in a different average policy. A brighter color in a tile indicates higher visitation frequency.



(a) No IM. (b) Max Entropy. (c) GRM + ME.

Figure 6: Empty final agent behavior example. Max Entropy (especially with GRM) results in a policy closer to optimality. A brighter color in a tile indicates higher visitation frequency.



(a) No IM. (b) State Count. (c) ICM.

Figure 7: Early exploration behavior on Empty.

and ICM performed better than the baseline, while Max Entropy did similarly or worse. On the other hand, performance on FourRooms was slightly better for the baseline model at the end of training. Based on reward performance alone, we know the final policies found by the IM agents deviated from the baseline.

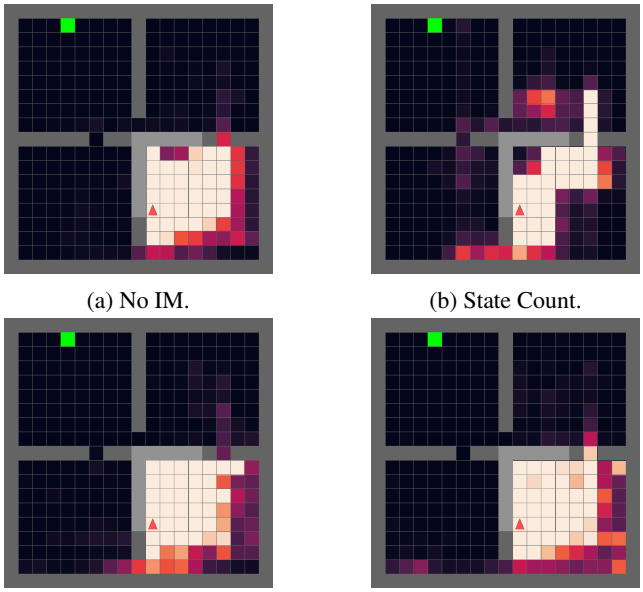
In terms of behavior analysis, we observed a similar trend of IM changing the final policy. The policy divergence metric (Table 3) shows that optimality is most preserved on the DoorKey and Empty maps, but less preserved on the larger FourRooms and RedBlueDoors maps. Figure 4 shows an example of State Count resulting in a different final policy than using no IM. Where the no IM agent favors a simple route of picking up the key from the top, the State Count model instead approaches the key from all surrounding positions: an instance of the noisy TV problem. We observed similar effects on Empty, where the agents sometimes route through the center of the map. Similarly, on FourRooms, as shown in Figure 5, the ICM and Max Entropy agents often fail to find the efficient path to the goal, through the bottom left, and instead favor the exploration of the rooms adjacent to the starting agent's.

Regardless of policy-altering behavior, these results highlight the benefits of IM methods. In terms of position coverage (First row of Figure 3), IM methods resulted in higher coverage in almost all cases, with the baseline agent not being able to bridge the gap on DoorKey and Empty. Moreover, while IM caused changes in the average reward obtained by the agent (First row of Figure 2), the change was positive for the State Count and ICM agents. Even in these simple MiniGrid environments, results show that IM agents found a better average policy than the baseline. For instance, the Max Entropy agent trained on Empty, shown in Figure 6, favors the optimal policy of going all the way right, then down to the goal[2]. Even then, the Max Entropy agent sometimes takes a slightly less optimal path (the slightly colored column beside the most frequent path), possibly due to the intrinsic reward.

Regarding the behavior of IM during early exploration, we generally observed changes in the positions frequented by the agent. Models trained with State Count frequented positions far away from the starting point or the optimal path compared to the baseline. On the other hand, Max Entropy and ICM models behaved quite similarly to the baseline. Figure 7 shows an instance of this behavior on Empty. In this scenario, State Count resulted in more frequent visitations of the spaces towards the center, whereas the baseline and other IM methods frequented the edges of the room. Models trained on the FourRooms map exhibited similar behavior, as shown in Figure 8, where State Count had higher state visitation counts on the rooms adjacent to the starting point. In contrast, ICM and State Count agents behave the opposite way on the DoorKey and RedBlueDoor maps, favoring instead the areas of space that lead to the extrinsic reward. It is likely that in these simpler maps, the agents had already gone through their initial intrinsic rewards and thus decreased exploration.

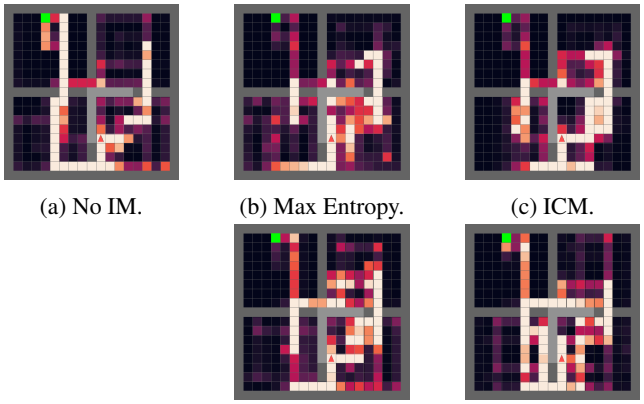
These results show that intrinsic motivation does change agent behavior, both during early training and on the fi-

[2]The path is optimal as turning requires an action on MiniGrid. Since the reward is inversely proportional to the number of actions needed to reach the goal, the shortest path is the top-right one since it requires only turning once.



(a) No IM. (b) State Count. (c) Max Entropy. (d) ICM.

Figure 8: Early exploration behavior on FourRooms.



(a) No IM. (b) Max Entropy. (c) ICM. (d) GRM+ME. (e) GRM+ICM.

Figure 9: FourRooms GRM agent behavior example. GRM policies resemble the baseline more closely than their no-GRM counterparts. A brighter color in a tile indicates higher visitation frequency.

nal policy. IM agents generally achieved earlier instances of the sparse reward and better early position coverage. In some cases, this initial advantage in metrics could never be bridged by the baseline. In terms of final policy, IM agents resulted in sufficiently different policies, both from a perspective of average reward (in all maps but Empty) and perceivable behavior. The impact of IM was mostly positive, and non-Max Entropy models attained better early exploration and more refined policies in all maps except FourRooms.

GRM Reduces Policy Divergence

The addition of GRM on top of the three studied intrinsic rewards slightly diminished the policy-altering effects of IM while retaining most of its benefits. In terms of average episodic reward (Figure 2, second to fourth rows), the GRM agents obtained reward values comparable to their counterparts, with near-equal performance in most cases. GRM decreased the return performance of the state count agent on FourRooms and RedBlueDoors, but increases it for Max Entropy (except on RedBlueDoors) and ICM. Notably, GRM made the agents on FourRooms have returns comparable to the better-performing baseline agent. In terms of position coverage, GRM slightly reduced it on the Empty map—which is positive since the map requires little exploration.

The greater effect of applying GRM was on the final policies of the agents. GRM generally lowered the policy divergence (Table 3) of Max Entropy and ICM on the more complex maps.

On Empty, GRM resulted in agents with policies closer to the optimal, except for State Count. Figure 6 shows the policies of these models, where the fully trained GRM with Max Entropy agent always chose the optimal path. We observe similar behavior on the DoorKey map, shown on Figure 4, where the agents trained with GRM resemble the no IM policy more closely than their counterparts, with more emphasis on the optimal path[3].

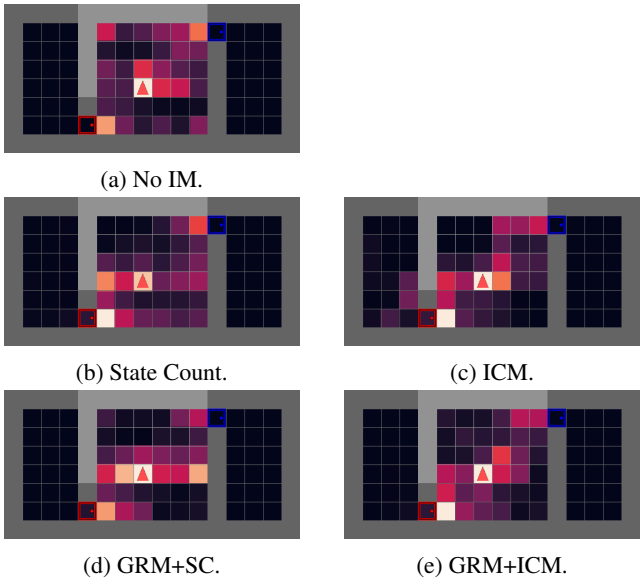
On the two complex maps, RedBlueDoors and FourRooms, GRM improved the performance of the non-State Count IM methods. On FourRooms, the two GRM models reach the goal tile with higher frequency, as shown in Figure 9, and their behavior (bottom row) more closely resembles the baseline policy when compared against their counterparts (top row, also shown on Figure 5). This observation is consistent with the calculated policy divergence (Table 3).

GRM hurt agent performance on the RedBlueDoors map. Firstly, it made State Count and ICM diverge further from the baseline policy, as shown in Table 3. In the case of State Count, it further decreased the average reward—although it slightly increased the performance of ICM. Figure 10 shows the behavior of these two GRM agents (bottom row) compared against their counterparts (middle row). The baseline behavior is erratic in this circumstance, although it somewhat resembles the optimal policy[4]. State Count policy favors the middle and lower parts of the room, while ICM took a very consolidated path. Adding GRM made it so agents take more erratic paths, staying within one tile of the middle paths that the base IM methods settled for. We note that these adverse effects might be specific to the shown map instance, although the divergence values indicate these effects are consistent.

Table 4 displays the average reward obtained on the final 100 training iterations. The results are consistent with our previous observations: GRM improved the performance of Max Entropy and ICM, but worsens that of State Count on the more complex maps. These results also highlight the power of GRM and ICM, which resulted in top performance

[3]The optimal path is to navigate to the tile above the key, grab it, move to and open the door, and move to the goal.

[4]Take either path to the red door, then move all the way up, then right towards the blue door.



(a) No IM.

(b) State Count.

(c) ICM.

(d) GRM+SC.

(e) GRM+ICM.

Figure 10: RedBlueDoors GRM agent behavior. A brighter color in a tile indicates higher visitation frequency.

	DK	EM	FR	RBD
No IM	0.7793	**0.9702**	**0.6119**	0.7813
SC	**0.9749**	**0.9737**	0.5966	**0.9578**
ME	0.7787	0.8785	0.4520	0.0668
ICM	**0.9743**	**0.9748**	0.5594	0.9458
GRM+SC	**0.9744**	**0.9747**	0.4742	0.6645
GRM+ME	0.8336	0.8857	**0.6119**	0.6838
GRM+ICM	**0.9745**	0.9184	**0.6023**	**0.9656**

Table 4: Average reward per-episode obtained during the last 100 training iterations, per map and model. Highlighted values are the highest per map or within 0.01.

on three out of four maps (tied with State Count), only having comparatively bad performance on the Empty map.

Lastly, we observed GRM to have little to no impact on early exploration. Agents trained with GRM still exhibited the behavior and obtained the benefits of IM.

The results highlight that GRM does reduce the policy divergence of IM in certain scenarios. Behavior exhibited by GRM agents resembles more closely what would be the optimal policy or that found by the baseline agent. We cannot state that GRM is a one-size-fits-all solution that grants all of the benefits of IM without any of the drawbacks. Not only did agents trained with GRM still result in policies far different from the no IM baseline, but it also worsened performance in some circumstances—particularly when combined with State Count.

Discussion

What is the behavior archetype of the IM methods? State Count methods result in more exploration during early training, and with good policies. Even during the latter stages of learning, these models have a stronger tendency to move off the beaten path. Max Entropy, on the other hand, tends to result in behavior that heavily favors a limited area of the space, resulting in a small number of well-traversed (lighter colored in the heatmap) grid positions. We observe that Max Entropy by itself scarcely finds early instances of the sparse reward, but it refines the final policy. It results in agents that act very risk-averse and prefer the 'safe' areas of the map. ICM works as a less extreme version of State Count, offering less early and late exploration.

Does IM truly result in sub-optimal policies? The results of this study mainly support that IM methods, especially with GRM, resulted in policies that we perceived to be closer to optimal. Longer training would likely result in the baseline model converging to the expected optimal policy. On the other hand, given enough time, it is likely that IM methods that run on diminishing rewards, such as State Count, will lead to the same result. An interesting follow-up question would then be how much training would each method need to reach optimality.

Are theoretical guarantees of optimality enough? GRM methods with theoretical guarantees on optimality resulted in policies that deviated from the baseline in small training horizons. While theoretically GRM guarantees an invariant policy, empirically, there might not be enough time or resources to achieve optimality.

What is optimality? Our initial setup employs PPO with no intrinsic motivation as our 'baseline' behavior, as it should eventually converge to the optimal policy. The results we obtained in this study were far from optimal, however, as we visualized a policy that was still very much in the middle of development. IM methods often performed better than the baseline, so they were 'better than optimal'. While this can be fixed by training for longer episodes, this paints a grim panorama for any analysis regarding behavior analysis and optimality. For instance, training an RL agent until it reaches an optimal policy might be unfeasible for sufficiently complex games. In even more complex games, such as Atari, the optimal policy is completely unknown to human or AI players. To address this issue, future work will have to consider non-reinforcement learning derived policies to have a consistent baseline behavior.

Is this type of behavior analysis enough? Much work is to be done in this regard. By including a baseline behavior observation, we are one step closer to talking about policy (in)variance for IM. These results raise some questions. Interpreting the heatmaps is not trivial and disregards the order of operations done by the agent, but switching to empirical video analysis has not been explored yet. Analysis of optimal behavior is harder since the optimal policy changes from map instance to instance, sometimes with multiple optimal policies. In addition to policy divergence, there may be other, undiscovered behavior-related metrics.

What is our recommended intrinsic motivation? We found State Count to be consistently effective at finding earlier instances of the sparse reward, plus it requires minimal setup. We note that GRM combined with ICM has potential, since it tended to 'smooth' over the final policy while retaining similar early exploration. While Max Entropy performed

poorly, it can be somewhat effective with GRM. Selection of an appropriate IM method requires proper characterization of the task complexity and computational resources.

Limitations

We identified a group of factors that could have potentially influenced the results of the experiment. All of these are current shortcomings that we will address in future work. We used a static set of hyperparameters for PPO, which we based on (Kayal, Pignatelli, and Toni 2025). We adjusted the value for the intrinsic reward coefficient β individually, but non-exhaustively. The implementation details of the PPO algorithm can affect its performance[5], which we mitigate by using the existing DEIR implementation.

Training Data: The models are trained for 10 million frames total, which we found was enough to find a stable policy in most cases. However, the possibility remains that with a longer training period, the performance of the models might change. We repeat the experiments a total of 10 times, which is double as many runs as the protocol study.

Metrics: For our return performance analysis, we used the metrics proposed in (Kayal, Pignatelli, and Toni 2025). For our behavior analysis, however, we rely on heatmaps and visual interpretation. To the best of our knowledge, there are no existing methods to analyze the behavior of reinforcement learning agents. In general, we report that behavior analysis is an understudied area of RL.

Generalizability: Grid-live environments of around the same complexity. Ideally, we would work with complex game environments such as Atari or MicroRTS, but instead chose MiniGrid as we expected behavioral analysis to be unfeasible on those games. Particularly, visualizing agent policies for those games is a challenge due to their complex state spaces. We will move on to actual game environments in our following studies.

Reproducibility: We have reported technical details of the experiment in an attempt to make these results replicable, and we have shared our source code and artifacts.

Conclusions

In this paper, we explore the effects of intrinsic motivation techniques through analysis of agent behavior across several variants. As a first step, we empirically analyze how three IM methods, plus GRM, change the behavior of agents trained on MiniGrid, in terms of return performance and observable behavior over training. Results indicate that IM is beneficial in most scenarios, but with varied observed in-game behavior of agents. There were both cases where reward hacking made the final policy of IM agents more and less optimal, hinting that the side effects of IM rewards might not be as undesirable as the literature suggests. While GRM has the theoretical guarantees of being policy-invariant, results showed it still creates behavior that deviates from the baseline for shorter training runs.

Our future work will extend the scope of this study to incorporate more complex environments, such as Atari Games and MicroRTS, more intrinsic motivation methods, including policy-invariant methods, and longer, repeated training. We will also research novel methods to better compare RL policies, as we found our current methods are rather limited to ascertain the policy (in)variance of any algorithm. We are also researching methods to generate a consistent, optimal policy to facilitate analysis. Behavior analysis on RL remains an underexplored area of literature, and we encourage researchers to join us in this challenge.

[5]See https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/.

Acknowledgements

The first author thanks the *University of Costa* Rica (*Universidad de Costa Rica*) for its contribution and support during his Ph.D. program.

References

Andres, A.; Villar-Rodriguez, E.; and Del Ser, J. 2022. An Evaluation Study of Intrinsic Motivation Techniques Applied to Reinforcement Learning over Hard Exploration Environments. In Holzinger, A.; Kieseberg, P.; Tjoa, A. M.; and Weippl, E., eds., *Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 201–220. Cham: Springer International Publishing. ISBN 978-3-031-14463-9.

Badia, A. P.; Sprechmann, P.; Vitvitskyi, A.; Guo, D.; Piot, B.; Kapturowski, S.; Tieleman, O.; Arjovsky, M.; Pritzel, A.; Bolt, A.; and Blundell, C. 2019. Never Give Up: Learning Directed Exploration Strategies. In *Proceedings of the Seventh International Conference on Learning Representations*.

Behboudian, P.; Satsangi, Y.; Taylor, M. E.; Harutyunyan, A.; and Bowling, M. 2022. Policy invariant explicit shaping: an efficient alternative to reward shaping. *Neural Computing and Applications*, 34(3): 1673–1686.

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; and Efros, A. A. 2018. Large-Scale Study of Curiosity-Driven Learning. In *Proceedings of the Eighth International Conference on Learning Representations*.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019. Exploration by random network distillation. In *Proceedings of the Seventh International Conference on Learning Representations*, 1–17.

Chen, E.; Hong, Z.-W.; Pajarinen, J.; and Agrawal, P. 2022. Redeeming intrinsic rewards via constrained optimization. *Advances in Neural Information Processing Systems*, 35: 4996–5008.

Chevalier-Boisvert, M.; Dai, B.; Towers, M.; Perez-Vicente, R.; Willems, L.; Lahlou, S.; Pal, S.; Castro, P. S.; and Terry, J. 2023. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *Advances in Neural Information Processing Systems*, 36: 73383–73394.

Colas, C.; Karch, T.; Sigaud, O.; and Oudeyer, P.-Y. 2022. Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey. *Journal of Artificial Intelligence Research*, 74: 1159–1199.

Forbes, G. C.; Gupta, N.; Villalobos-Arias, L.; Potts, C. M.; Jhala, A.; and Roberts, D. L. 2024a. Potential-Based Reward Shaping for Intrinsic Motivation. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 589–597. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 979-8-4007-0486-4.

Forbes, G. C.; Villalobos-Arias, L.; Wang, J.; Jhala, A.; and Roberts, D. L. 2024b. Potential-Based Intrinsic Motivation: Preserving Optimality With Complex, Non-Markovian Shaping Rewards. ArXiv:2410.12197 [cs].

Forbes, G. C.; Wang, J.; Villalobos-Arias, L.; Jhala, A.; and Roberts, D. L. 2025. Action-Dependent Optimality-Preserving Reward Shaping. ArXiv:2505.12611 [cs].

Huang, S.; and Ontañón, S. 2020. Action Guidance: Getting the Best of Sparse Rewards and Shaped Rewards for Real-time Strategy Games. ArXiv:2010.03956 [cs].

Kayal, A.; Pignatelli, E.; and Toni, L. 2025. The impact of intrinsic rewards on exploration in Reinforcement Learning. *Neural Computing and Applications*, 37(21): 16269–16303.

Ladosz, P.; Weng, L.; Kim, M.; and Oh, H. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85: 1–22.

Laskin, M.; Yarats, D.; Liu, H.; Lee, K.; Zhan, A.; Lu, K.; Cang, C.; Pinto, L.; and Abbeel, P. 2021. URLB: Unsupervised Reinforcement Learning Benchmark. In *Proceedings of the Deep RL Workshop NeurIPS 2021*.

Le, H.; Do, K.; Nguyen, D.; and Venkatesh, S. 2024. Beyond Surprise: Improving Exploration Through Surprise Novelty. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 1084–1092. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 979-8-4007-0486-4.

Liu, J.; Gu, X.; and Liu, S. 2020. Policy Optimization Reinforcement Learning with Entropy Regularization. ArXiv:1912.01557 [cs].

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with Deep Reinforcement Learning. ArXiv:1312.5602 [cs].

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533. Publisher: Nature Publishing Group.

Mohamed, S.; and Jimenez Rezende, D. 2015. Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning. In *Advances in Neural Information Processing Systems 28*, volume 28. Curran Associates, Inc.

Ontañón, S.; Barriga, N. A.; Silva, C. R.; Moraes, R. O.; and Lelis, L. H. S. 2018. The First microRTS Artificial Intelligence Competition. *AI Magazine*, 39(1): 75–83.

Oudeyer, P.-Y.; and Kaplan, F. 2007. What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, 1. Publisher: Frontiers.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In *Proceedings of the 34th International Conference on Machine Learning*, 2778–2787. PMLR. ISSN: 2640-3498.

Raileanu, R.; and Rocktäschel, T. 2019. RIDE: Rewarding Impact-Driven Exploration for Procedurally-Generated Environments. In *Proceedings of the Eighth International Conference on Learning Representations*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. ArXiv:1707.06347 [cs].

Strehl, A. L.; and Littman, M. L. 2008. An analysis of model-based Interval Estimation for Markov Decision Processes. *Journal of Computer and System Sciences*, 74(8): 1309–1331.

Taiga, A. A.; Fedus, W.; Machado, M. C.; Courville, A.; and Bellemare, M. G. 2019. On Bonus Based Exploration Methods In The Arcade Learning Environment. In *Proceedings of the Eighth International Conference on Learning Representations*.

Taïga, A. A.; Fedus, W.; Machado, M. C.; Courville, A.; and Bellemare, M. G. 2021. Benchmarking Bonus-Based Exploration Methods on the Arcade Learning Environment. ArXiv:1908.02388 [cs].

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; Oh, J.; Horgan, D.; Kroiss, M.; Danihelka, I.; Huang, A.; Sifre, L.; Cai, T.; Agapiou, J. P.; Jaderberg, M.; Vezhnevets, A. S.; Leblond, R.; Pohlen, T.; Dalibard, V.; Budden, D.; Sulsky, Y.; Molloy, J.; Paine, T. L.; Gulcehre, C.; Wang, Z.; Pfaff, T.; Wu, Y.; Ring, R.; Yogatama, D.; Wünsch, D.; McKinney, K.; Smith, O.; Schaul, T.; Lillicrap, T.; Kavukcuoglu, K.; Hassabis, D.; Apps, C.; and Silver, D. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354. Publisher: Nature Publishing Group.

Wan, S.; Tang, Y.; Tian, Y.; and Kaneko, T. 2023. DEIR: efficient and robust exploration through discriminative-model-based episodic intrinsic rewards. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, 4289–4298. Macao, P.R.China. ISBN 978-1-956792-03-4.

Zhang, T.; Xu, H.; Wang, X.; Wu, Y.; Keutzer, K.; Gonzalez, J. E.; and Tian, Y. 2021. NovelD: A Simple yet Effective Exploration Criterion. In *Advances in Neural Information Processing Systems 34*, volume 34, 25217–25230. Curran Associates, Inc.