Characteristics of Effective Exploration for Transfer in Reinforcement Learning

Jonathan C. Balloch
balloch@gatech.edu
College of Computing
Georgia Institute of Technology

Rishav Bhagat
rishavbhagat.cs@gmail.com
College of Computing
Georgia Institute of Technology

Geigh Zollicoffer
gzollicoffer3@gatech.edu
College of Computing
Georgia Institute of Technology

Ruoran Jia
rjia41@gatech.edu
College of Computing
Georgia Institute of Technology

Julia M. Kim
julia.kim@gatech.edu
College of Computing
Georgia Institute of Technology

Mark O. Riedl
riedl@cc.gatech.edu
College of Computing
Georgia Institute of Technology

Abstract

In reinforcement learning (RL), exploration is used to help policy models learn to solve individual tasks more efficiently and in increasingly challenging environments. In many real-world applications of RL, however, environments are non-stationary; they can change in unanticipated and unanticipatable ways, and there are conditions in which the agent must adapt its policy *online*, at test time, to the changed environment. Given that most exploration methods are designed for stationary MDPs of single tasks, it is not well understood which exploration methods are most beneficial to efficient online task transfer. Our first contribution is to categorize an array of exploration methods according to common "characteristics" such as being designed around the principles of a separate exploration objective or adding noise to the RL process. We then evaluated eleven exploration algorithms within and across characteristics on the efficiency of adaptation and transfer in multiple discrete and continuous domains. Our results show that exploration methods designed around the principle of *explicit diversity* and *stochasticity* most consistently benefit policy transfer. Additionally, our analysis considers the reasons that some characteristics correlate with improved performance and efficiency across multiple tasks, while others only improve transfer performance with respect to specific tasks. We conclude by discussing the potential implications for future exploration algorithms to most efficiently adapt to unexpected test-time environment changes.

1 Introduction

Modeling the world as a *stationary* Markov decision process has been fundamental to understanding and advancing reinforcement learning (RL). However, many real-world problems used to motivate RL research, such as robotics (Ibarz et al., 2021), autonomous driving (Kiran et al., 2021), power distribution (Zhang et al., 2019), and language preferences (Casper et al., 2023), are in fact *non-stationary*. Robust models can accommodate some amount of non-stationarity, for example in the form of noisy sensor observations and/or state transitions. However, an agent may also experience *novelties*, which are sudden, permanent changes to the observation space or environment state

transition dynamics that occur during deployment that are unanticipated—and unanticipatable—by the agent because they are not observed during training and cannot be handled by robustness (Boult et al., 2021; Balloch et al., 2022; Muhammad et al., 2021).

When there is a novel change, the RL agent's converged policy can become ineffective or even make catastrophic and harmful mistakes, and as such require adaptation. One way to approach adaptation in RL is to transfer knowledge from the prior environment to the new environment—known as task transfer or transfer learning (Taylor & Stone, 2009). Transfer learning typically employs a period in which the agent can train in the new environment prior to further inference. However, in many real-world scenarios, the agent may not have the opportunity to train offline and instead must adapt its policy online (Zhan & Taylor, 2015). In this work, we specifically address the challenge of *online task transfer*, where an unanticipated, permanent shift in environment requires test-time policy performance recovery.

Reinforcement learning algorithms address the exploration-exploitation trade-off differently. Overall, the exploration phase is necessary to sample the state-action space in order for the policy model to learn to solve an individual task more efficiently. In theory, exploration designed for stationary RL can enable agents to experience and adapt to environment novelties with no fundamental changes (Schmidhuber, 1991a;b; Chentanez et al., 2004). Mirroring findings in biological animal behavior (Réale et al., 2007), prior work has found that exploration can improve the RL agent's performance and efficiency in transfer (Taylor & Stone, 2009; Silver et al., 2013; Langley, 2022). In spite of this, exploration algorithms designed to improve the exploration-exploitation trade-off of solving single, stationary MDPs have not been comprehensively analyzed for their impact on efficient online task transfer.

In this research, we answer the question: **which characteristics of traditional exploration algorithms are important for efficient transfer in RL?** We conducted experiments with eleven popular RL algorithms on five novelties in discrete and continuous domains. The algorithms were selected to represent a diverse space of exploration characteristics. We systematically examine the within- and between-class relationships of the algorithms across all characteristics. Our results indicate, foremost, that exploration methods that explicitly emphasize *diverse training experiences* and use *stochasticity* to avoid overfitting benefit policy transfer the most. This is true across all types of novelties and for both discrete and continuous domains. When novelty makes a task easier—called a *shortcut novelty*—, exploration methods that rely heavily on stochasticity lose some effectiveness, but the benefits of diversity are more pronounced. When novelty makes the task harder—called a *barrier novelty*—we that the difference in performance between all exploration methods was severely diminished. Finally, our continuous control experiments showed even more pronounced benefit of stochasticity and that exploration methods that are time independent or explore based on the entire training process—*temporally global* methods—outperformed methods that explore based on short-term change.

In this paper, we begin with preliminaries on reinforcement learning and online task transfer. Section 3 defines the exploration characteristics that have observable effects on transfer and maps eleven RL algorithms chosen for the experiment to these characteristics. Section 4 details our experimental methodology. Section 5 details our results and discusses implications.

2 Preliminaries

Reinforcement learning typically models an environment as a stationary Markov decision process (MDP): $M = \langle \mathcal{S}, \mathcal{A}, R, P, \gamma \rangle$, where $\mathcal{S}$ is the space of environment states, $\mathcal{A}$ is the space of actions, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the function that maps states and actions to a scalar reward, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function between states, and γ is the discount factor of future reward (Sutton & Barto, 2018). The learning task in RL is to learn a policy $\pi(a|s)$ that, for a given state, selects the action that maximizes the expectation of discounted future rewards. Critically, as reinforcement learning is typically formulated, agents cannot only greedily pursue future reward; to find an optimal policy they must trade off exploration with exploitation (Sutton & Barto, 2018). Exploration also

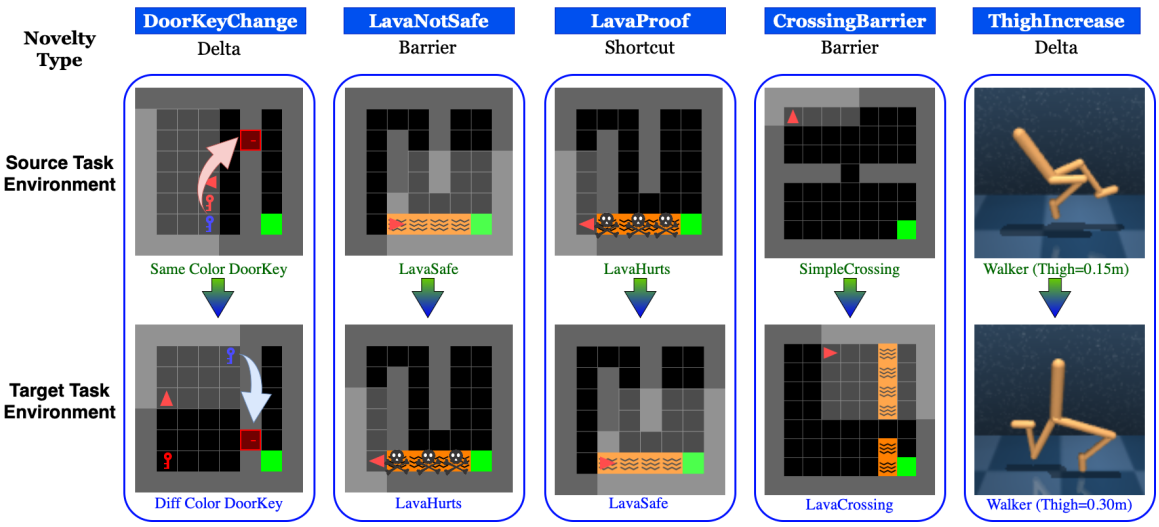

Figure 1: Environments and novelties used to evaluate the exploration algorithms and their characteristics, including discrete and continuous control environments.

copes with non-smooth learning challenges such as local minima and credit assignment challenges that result from sparse or time-varying rewards, which has been shown to also improve maximum performance especially in dense, unbounded reward settings like Atari video games (Taiga et al., 2020).

Online Task Transfer is a subdomain of transfer learning, wherein a parameterized model seeks to adapt knowledge from a prior problem to improve training on a new problem. However, transferring knowledge can cause parameterized models to experience catastrophic inference (McCloskey & Cohen, 1989) where the training data from the new distribution cause the agent to transfer little, if any, of its previous model when learning the new task (Zhu et al., 2023). We model the transfer process for deep reinforcement learning as a transition from MDP_{source} and MDP_{target}, mirroring the differentiation into source and target tasks in supervised transfer learning (Zhu et al., 2023). In online task transfer (Zhan & Taylor, 2015), RL agents trained on a MDP_{source} must adapt their policy to solve MDP_{target} online—at test time—while being evaluated in that new target MDP. The "injection" of the novelty separates the source and target tasks. Balloch et al. (2022) provides a taxonomy with which to analyze novelties that distinguish the source and target tasks; they divide novelties into *shortcut*, *barrier*, and *delta* novelties. A novelty is a *shortcut* when the target task is less complex than the source task, a *barrier* when the target task is more complex than the source task, and a *delta* when the source and target tasks have similar independent complexity. In this work, we characterize changes in the environment as one of these three types of novelty.

3 Characterizing Exploration Methods

There are many ways one might categorize exploration methods. Our experiments indicate that, from the perspective of OTT, exploration methods can be split into two high-level categories: *exploration principle* and *temporal locality*, which both have subcategories. These are consistent with the existing taxonomy of Ladosz et al. (2022).

Exploration principle characterizes an agent's behavior beyond greedy maximization of reward. We identified three categories of exploration principles. (1) Adding *stochasticity* into the learning process. There are many ways to use stochasticity in exploration, whether by injecting random noise into the input or an intermediate weight layer, using a stochastic task policy, or simply selecting random actions. (2) *Explicit diversity* over the different random variables in the process. Explicit

Categories	Characteristics	Example Algorithms
Exploration Principle	Stochasticity	NoisyNets, DIAYN
	Explicit Diversity	RND, REVD, RISE, RE3, RIDE, NGU, DIAYN
	Separate Objective	RND, RIDE, ICM, NGU, GIRL
Temporal Locality	Global	RND, ICM, RE3, NGU, GIRL
	Local	EVD, RIS, RIDE, NGU
	Time Independent	NoisyNets, DIAYN

Table 1: This table lays out our decomposition of exploration algorithms into two major categories—exploration principle and temporal locality—with three core characteristics in each. The algorithms listed here are evaluated as described in Section 4.1. Algorithms are described in detail in the Appendix.

diversity methods encourage models to experience all parts of the domain and task equally, ensuring that a greedy process does not lead the agent into stale transitions. (3) Having a *separate objective* in addition to greedy pursuit of reward. Methods with a separate objective complement the flaws of greedy reward maximization with a non-greedy goal, alternating or combining the objectives.

Temporal locality characterizes an exploration algorithm's relationship to time. Most exploration algorithms are designed to adapt to the needs of an agent at different points in the learning process. We identified three temporal locality categories. (1) Algorithms with short-term or temporally *local* characteristics. These methods implement adaptive behavior as a function of how agent and environment properties evolve time step to time step or episode to episode. (2) Algorithms with long-term temporally *global* characteristics. These methods influence exploration based on trends in agent and environment properties recorded or aggregated across the entire learning problem or by comparing these global properties with the current agent, environment, or learning state. (3) *Time-independent* exploration methods. Similar to characterizations (Sutton & Barto, 2018) of exploration methods as "directed" or "undirected," time-independent methods counteract greedy behavior by altering the learning process as a whole or within the agent architecture itself. Time-independent methods are critical to evaluation of exploration in transfer applications because online task transfer induces a temporal shift, both globally and locally. We summarize the exploration principle and temporal locality categories, along with exemplar algorithms, in Table 1 and Appendix B.

4 Experiments

We selected 11 reinforcement learning algorithms based on their exemplary usage of stochasticity, explicit diversity, separate exploration objectives, and orientation to global or local temporal locality. We trained and tested each algorithm in discrete and continuous domains and in the presence of shortcut, delta, or barrier novelties.

4.1 Exploration Algorithms

For our assessment, we focus on model-free, on-policy deep policy gradient methods that apply to a variety of reinforcement learning tasks. Specifically, we use proximal policy optimization (PPO) (Schulman et al., 2017), a high-performing actor-critic policy gradient method, as the algorithmic backbone of all the exploration methods we test. On-policy actor-critic methods such as PPO are more versatile than off-policy methods, which only apply to a subset of RL problem formulations. For example, methods like Deep Q-Networks (Mnih et al., 2015) only apply to problems with discrete action spaces, and methods like Soft-Actor Critic (Haarnoja et al., 2018) and Deep Deterministic Policy Gradients (Lillicrap et al., 2019) only work in continuous control environments. Additionally, off-policy methods are very sensitive to the management of an experience replay buffer for successful learning (Mnih et al., 2015), which becomes significantly more complex when adapting online because hyperparameters such as how often the experience replay buffer should be reset become potential confounding variables. In an effort to control as many independent variables as

Figure 2: Full learning and adaptation process of eleven RL exploration algorithms on the `DoorKeyChange` novelty problem from NovGrid (Balloch et al., 2022). The agents first learn a task assuming a stationary MDP. The rate of learning at this stage is *convergence efficiency*. At time step 5,000,000 novelty is injected into the environment, transferring from MDP_{source} to MDP_{target}, often causing a performance drop-off. The algorithms then recover their performance as they learn the new world transition dynamics. The rate of learning at this stage is *adaptive efficiency*. The maximum episode reward is the *final adaptive performance*, which may not always be as high as pre-novelty performance.

possible and focus our investigation on exploration, we only consider the PPO algorithm for this initial investigation.

We selected 11 popular exploration algorithms that represent a broad sampling of exploration principle and temporal locality categories, while being compatible with PPO and our environments. Those algorithms are Random Network Distillation (RND) (Burda et al., 2018b), Intrinsic Curiosity Module (ICM) (Pathak et al., 2017), Never Give Up (NGU) (Badia et al., 2020), Rewarding Impact-Driven Exploration (RIDE) (Raileanu & Rocktäschel, 2019), Renyi State Entropy Maximization (RISE) (Yuan et al., 2022b), Rewarding Episodic Visitation Discrepancy (REVD) (Yuan et al., 2022a), Generative Intrinsic Reward Learning (GIRL) (Yu et al., 2020), Parameter Space Noise for Exploration (NoisyNets) (Plappert et al., 2018), and "online" Diversity Is All You Need (DIAYN) (Eysenbach et al., 2019). Table 1 shows how the algorithms relate to exploration characteristics; descriptions of the algorithms can be found in Appendix A. Our implementation of these algorithms is based on the Stable-Baselines3 (Raffin et al., 2021) and RLeXplore libraries,[1] which we modify and expand for the purposes of our investigation.

4.2 Learning Environments and Transfer Tasks

To experiment with online transfer, agents are trained to convergence in one environment (the source task), and then a novelty is introduced to create the target task. The agent must recover its performance during online execution in the target environment. We run our experiments with two transfer learning libraries, NovGrid (Balloch et al., 2022) and Real-World Reinforcement Learning suite (Dulac-Arnold et al., 2021).

NovGrid is a specialization of the MiniGrid (Chevalier-Boisvert et al., 2023) environment designed to promote experimentation in novelty adaptation in RL. Specifically, NovGrid sets up learning scenarios and then injects a novelty—changing the transition dynamics—at a time that is unknown to the agent. We use three novelty environments within NovGrid—`DoorKey`, `LavaMaze`, and `CrossingBarrier` environment—which are used with the injection of specific nov-

[1] https://github.com/RLE-Foundation/RLeXplore

elties. `DoorKeyChange` is a delta novelty in which a DoorKey environment is changed so that the key that opens the door is changed. `LavaProof` is a shortcut novelty where the lava in LavaMaze is changed from being a zero-reward terminal state into a safe, passable, non-terminal state. `LavaNotSafe` is a barier novelty that is functionally the reverse of `LavaProof`, changing the lava in `LavaMaze` from non-terminal into a terminal state. Lastly, in `CrossingBarrier`, the impassable but safe walls are exchanged for standard, terminal-state lava. We allowed the algorithms to run until the majority of runs on all algorithms converged before the novelty was injected. We tuned the hyperparameters of the algorithms on the novelty-free `DoorKey` environment for use with the NovGrid environments, maximizing convergence in the source environment to help ensure convergence on the source task. The details of the hyperparameter tuning are in Appendix E.1.

The Real-World Reinforcement Learning suite (Dulac-Arnold et al., 2021) provided a continuous control environment to evaluate adaptation performance. We tuned the hyperparameters of our algorithms in the Cartpole-Swingup environment by changing the pole length, which maintains the same approximate difficulty of the target task. We evaluated OTT performance on the more complex Walker2D environment task with the novelty ThighIncrease, where the length of the thigh link is increased from 0.15 meters to 0.3 meters. See Figure 1 for illustrations of the environments and novelties.

4.3 Metrics for Online Task Transfer

To assess the exploration methods, we measure learning efficiency and performance motivated by the desire to minimize the number of environment interactions required to learn good policies in the target task. The primary metrics are the following.

Adaptive efficiency: The number of environment steps necessary for the agent to reach 95% of maximum performance on the target task.

Transfer Area Under the Curve (Tr-AUC): Inspired by the performance ratio of Taylor & Stone (2009), Tr-AUC is a novelty-agnostic measure of the overall transfer performance as a function of both the source and target tasks:

$$\text{Tr-AUC} = \frac{1}{2}\left(\max(r_{\text{S}}) + \frac{1}{K}\sum_{i \in K} r_{i,\text{T}}\right)$$

where $\max(r_{\text{S}})$ refers to the final performance on the source task and the summation over r_{T} gives accumulated adaptive performance until the final adaptive performance point on the target task. Tr-AUC balances efficient adaptation with prior task performance by penalizing methods that performed well on the target task due to underperforming on the source task or vice versa. For all metrics, we calculate the mean and standard deviation of a bootstrapped sampling of the runs of each method, and calculate the interquartile mean (IQM) and the bootstrapped 95% confidence interval per Agarwal et al. (2021).

One of the key assumptions that we make in the motivation of this work is that in real-world online task transfer scenarios, the policy is assumed to have converged to maximize the performance on the source task before novelty is injected, and the policy must be adapted to the target task. However, in practice, one of the deficiencies of deep reinforcement learning is the highly stochastic nature of convergence, especially in sparse reward tasks like those of NovGrid. For an analysis best aligned with our motivations, we measure our results with respect to the full set of experiments that converged on the source task unless otherwise specified.

5 Results and Discussion

We compared the relationship between source task convergence efficiency with adaptive efficiency for different algorithms in our environments and validated our analysis of these comparisons with results on the Tr-AUC metrics, exemplified in Figure 5. A complete list of our results for all algorithms, metrics, environments, and novelties can be found in Appendix D. We discuss and analyze our results in the context of the specific experimental research questions presented in Section 4.

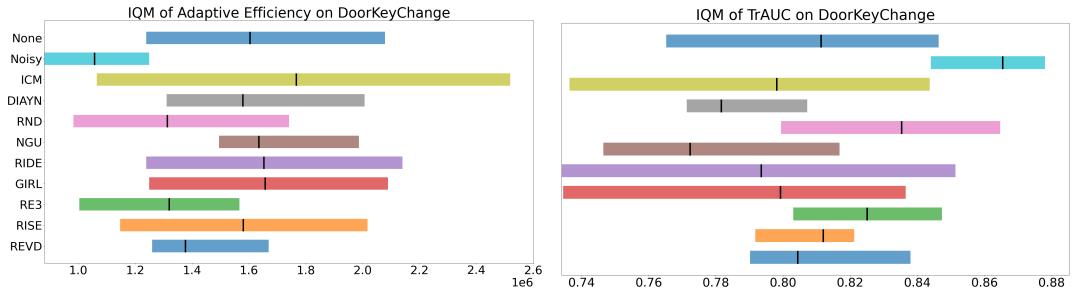

Figure 3: The Adaptive Efficiency and Tr-AUC inter-quartile mean plots for DoorKeyChange. These plots show that NoisyNets perform well in both metrics. It should be noted that the Adaptive Efficiency graphs are only showing runs that converged on both tasks and the Tr-AUC graphs are filtering for runs that converged on the first task.

The exploration principle characteristics have a large impact on the effectiveness of online task transfer. Exploration methods with *stochasticity* and *explicit diversity* characteristics are slower to converge on the source task, but adapt most efficiently to the target task. Representing the exploration principles of explicit diversity and stochasticity, respectively, RE3 and NoisyNets are the two algorithms that consistently performed well. Although not as consistent in performance as RE3 and NoisyNets, other explicit diversity and stochasticity methods REVD, RND, and DIAYN also adapt efficiently in most tasks, as can be seen in Tables 2 and 3.

Further reflecting the importance of exploration principle, ICM, NGU, and other *separate objective* performed consistently below average on the tasks and novelties. This can be attributed to inductive bias caused by the task-dependence of separate objective exploration methods. The ICM exploration method adds an inductive bias to the typical prediction error-based curiosity metric by focusing only on state change predictions *that result from agent action*. This is a productive approach in conventional single-task RL because it is robust to arbitrary changes in the environment, like the "Noisy TV" problem (Burda et al., 2018b). However, in online task transfer this would mean the exploration algorithm might avoid the novelty as it was not caused by agent action. NGU and several other separate objective algorithms use a similar action-focused inductive bias in their embedding spaces and, as a result, also see their performance suffer.

In continuous action environments, exploration methods with stochastic principles dominate, and the difference between of temporal locality characteristics is more important than in discrete action environments. Stochastic methods dominate in the continuous action domain. DIAYN and NoisyNets recover significantly faster than all other methods. Diversity in exploration for transfer is less important to efficiency than in discrete experiments but still performs on par with the non-stochastic exploration methods. This is most likely a result of higher transferability of the continuous control skills learned in the ThighIncrease novelty compared to the discrete environments novelties; because of the nature of a continuous action space, noise in both the random conditioning space of DIAYN and the noisy weights of NoisyNets exposes those policies to "nearby" actions corresponding to the new optimal policy. The explicit diversity principles underlying methods like RE3, REVD, and RND are more impactful in discrete action space environments as the optimal actions in MDP_{target} are not similar to the optimal actions in MDP_{source}. That said, diversity methods are not significantly worse in adaptive performance than separate objective methods, thus remain useful.

Temporal locality showed greater impact on performance in our continuous environment compared to our discrete environment. As shown in Figure 4, we find that the time-independent strategies—NoisyNets and DIAYN—dominate; the temporally global strategies such as RND, ICM, and NGU perform well; and the temporally local strategies struggle the most both pre- and post-novelty. We attribute this result to optimal continuous control policies often only needing small, smooth

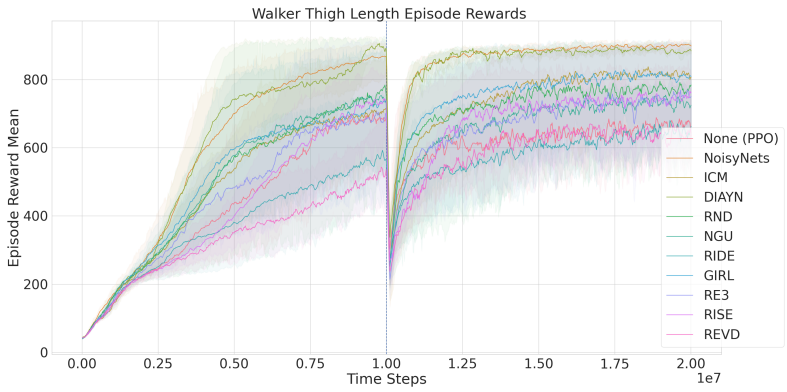

Figure 4: The reward plot from dm_control Walker-Walk ThighIncrease delta novelty transfer task. The vertical line at 1E7 steps indicates where novelty was injected. The shaded areas represent the variance over all seeds. NoisyNets and DIAYN are the highest performing and most efficient adapting methods. In contrast to the DoorKeyChange discrete delta novelty, there appears to be some correlation between performance before and after the novelty. The shaded areas represent the variance over all seeds.

Figure 5: Results from the `LavaSafe` shortcut novelty. The vertical line at 1E7 steps indicates where novelty was injected. The shaded areas represent the variance over all seeds. Some of the exploration algorithms are able to find the shortcut, rising above the pre-novelty performance, while others never discover the shortcut.

action differences in time to learn a policy, which favors exploration methods with global and time-independent temporal locality.

Compared to delta novelties, shortcut novelties increase the importance of diversity principles, while barrier novelties demonstrate the limitations of exploration to improve transfer in general. On delta novelties DoorKeyChange and ThighIncrease, stochastic methods have very good general performance. However, in the LavaProof shortcut novelty, the stochastic method NoisyNets fails to adapt and find the shortcut novelty, whereas the DIAYN stochastic method excelled. DIAYN differentiates itself from NoisyNets by combining elements of stochasticity with explicit diversity. As can be seen in Figure 5, globally temporal methods NGU, GIRL, and ICM also fail to consistently identify the shortcut over the safe lava in spite of learning how to safely navigate around it in the source task. One possible explanation is that shortcut novelties have no

Exploration Algorithm	Adaptive Efficiency ↓				
	DoorKeyChange (10^6)	LavaNotSafe (10^6)	LavaProof (10^4)	CrossingBarrier (10^5)	ThighIncrease (10^6)
None (PPO)	1.5 ± 0.477	2.56 ± 2.09	**2.05 ± 0.0**	6.48 ± 3.15	3.4 ± 2.51
NoisyNets	**0.965 ± 0.204**	0.963 ± 0.534	7.58 ± 9.15	5.88 ± 3.72	1.69 ± 0.538
ICM	1.57 ± 0.589	7.58 ± 1.26	**2.05 ± 0.0**	7.69 ± 4.69	4.18 ± 1.39
DIAYN	1.52 ± 0.422	3.65 ± 2.47	5.8 ± 5.31	5.43 ± 3.71	**1.66 ± 0.389**
RND	1.23 ± 0.385	4.64 ± 3.63	**2.05 ± 0.0**	5.25 ± 2.39	2.81 ± 1.46
NGU	1.58 ± 0.317	2.39 ± 1.38	6.4 ± 11.5	4.41 ± 4.02	3.71 ± 1.74
RIDE	1.53 ± 0.527	4.51 ± 2.34	2.56 ± 1.35	5.32 ± 3.71	5.18 ± 2.73
GIRL	1.57 ± 0.541	5.49 ± 3.1	**2.05 ± 0.0**	6.31 ± 4.57	3.08 ± 1.98
RE3	1.21 ± 0.312	**0.896 ± 0.21**	**2.05 ± 0.0**	**4.14 ± 2.07**	4.32 ± 1.81
RISE	1.41 ± 0.374	1.37 ± 0.478	**2.05 ± 0.0**	4.67 ± 3.26	3.6 ± 0.597
REVD	1.27 ± 0.319	2.43 ± 1.34	2.87 ± 1.64	5.43 ± 3.24	3.92 ± 0.202

Table 2: This table shows the mean and variance of the adaptive efficiency on the post-novelty tasks. It is computed by calculating the number of steps from the start of the novel task until convergence on the second task. Thus, lower numbers are better. Only runs that converged on both tasks are taken into account for this metric.

Exploration Algorithm	Transfer Area Under Curve ↑				
	DoorKeyChange (10^{-1})	LavaNotSafe (10^{-1})	LavaProof (10^{-1})	CrossingBarrier (10^{-1})	ThighIncrease (10^2)
None (PPO)	7.72 ± 0.792	7.43 ± 1.29	9.66 ± 0.0835	8.89 ± 0.297	6.5 ± 1.63
NoisyNets	**8.13 ± 1.23**	**8.37 ± 0.885**	7.69 ± 3.36	8.94 ± 0.388	8.62 ± 0.39
ICM	7.28 ± 1.07	5.43 ± 0.667	9.22 ± 1.16	8.74 ± 0.537	7.25 ± 1.56
DIAYN	7.54 ± 0.624	6.25 ± 1.22	**9.7 ± 0.0773**	9.01 ± 0.493	**8.72 ± 0.203**
RND	8.09 ± 0.542	6.25 ± 1.53	9.37 ± 0.66	9.0 ± 0.399	7.47 ± 1.73
NGU	7.56 ± 0.508	6.86 ± 1.38	9.48 ± 0.38	9.09 ± 0.444	7.07 ± 1.68
RIDE	7.67 ± 0.727	7.63 ± 0.895	9.5 ± 0.605	9.02 ± 0.373	5.76 ± 1.67
GIRL	7.59 ± 0.855	6.01 ± 1.08	9.55 ± 0.295	8.86 ± 0.51	7.45 ± 1.74
RE3	8.12 ± 0.387	6.82 ± 1.48	9.37 ± 0.524	**9.1 ± 0.266**	6.77 ± 1.99
RISE	7.35 ± 1.08	7.07 ± 1.77	9.42 ± 0.402	9.09 ± 0.343	7.05 ± 1.03
REVD	7.99 ± 0.402	7.3 ± 1.68	9.69 ± 0.056	8.92 ± 0.384	5.58 ± 1.33

Table 3: The mean and variance of the transfer area under the curve metric, which is computed by adding final reward on the first task with the area under the reward curve in the second task. Higher numbers indicate better adaptation. This only includes runs that converged on the first task.

performance drop that forces models to explore more. In that case, it illustrates a scenario in which implementation of an exploration principle would be very important, such as to require a principle of explicit diversity.

On the other extreme, the barrier novelty results from CrossingBarrier and LavaNotSafe showed methods with exploration principles of stochastic and explicit diversity generally continued to be most effective, but there is larger variance between methods within and across the categories. This difference in variance is especially obvious in the CrossingBarrier task, where adaptive efficiency and Tr-AUC variances are as high as 91.1% of the mean. These findings suggest the limits of exploration to improve transfer. For the barrier novelties, there is the target task solution is significantly longer than the target task solution compared to barrier novelties, meaning that often less prior knowledge can be transferred. Thus, at the extreme, online task transfer for a barrier novelty is akin to learning two single-tasks with no prior knowledge, as compared to online task transfer for a delta or shortcut novelty. It illuminates online task transfer's implicit assumption that some knowledge learned in the source task can be transferred to the target task, and suggests that most general purpose exploration

methods, such as those studied in this work, are unlikely to benefit policy adaptation to difficult barrier novelties in general.

6 Related Work

There is a large body of work characterizing and surveying the impact of exploration on transfer in RL. These works consider transfer in RL where exploration is a single variable (Taylor & Stone, 2009; Lazaric, 2012; Da Silva & Costa, 2019; Zhao et al., 2020; Zhu et al., 2023) and exploration as one of several use cases (Ladosz et al., 2022; Yang et al., 2021). There is also a body of work that examines the relationship between active learning and adaptation to novelty and open-worlds (Langley, 2020; Boult et al., 2021). Our work contributes by characterizing exploration methods across multiple dimensions and analyze their transfer performance specifically for RL and sequential decision-making.

Of the techniques investigating exploration methods for transfer in RL, they are tailored to a specific algorithm (Zhan & Taylor, 2015; Barreto et al., 2017), do not translate to deep RL (Konidaris et al., 2012), or do not compare themselves to stationary MDP exploration methods. Our work contributes by providing new analytical frameworks for further developing exploration methods depending on the transfer problem. Most similar to our work is (Burda et al., 2018a), which empirically investigates the implications of different exploration algorithms that share a curiosity objective as their exploration principle. Our work distinguishes itself by including a broader group of exploration principles than just intrinsic reward and does so for the purposes of online task transfer in RL instead of the typical single-task formulation.

7 Conclusions

In this work, we evaluated several deep reinforcement learning exploration algorithms on a number of online task transfer problems. Our results and analysis reveal four key findings: (1) Exploration principles of *explicit diversity*, represented by a method such as RE3, and *stochasticity*, such as NoisyNets, are the most consistently positive exploration characteristics across our novelty and environment types. (2) Time-independent and stochasticity-based exploration methods are best suited to online task transfer in the continuous control tasks, whereas temporal locality characteristics are less important in discrete control tasks. (3) The relative importance of exploration characteristics like explicit diversity varies with novelty type.

These results can help developers of RL systems understand how their choice of exploration algorithm could affect downstream performance in the face of novelties and online task transfer. They also point to areas where research can have greater impact as reinforcement learning matures to increasingly complex environments with real-world characteristics like novelty and require inference time adaptation. This includes the possibility of more exploration techniques like DIAYN that combine multiple exploration principles in a single method.

Broader Impact Statement

As reinforcement learning finds broader, real world applications, we as a research community must understand that improvements for algorithms both increase the likelihood of positive adoption and misuse. As this work is motivated by the real-world problem of non-stationary environment adaptation, we believe that our work has the potential for increasing adoption of both positive and negative applications. However, we believe the risk from this research is low: our work, while impactful, has a low technical readiness level, and there are many more steps necessary before the results here can be applied to critical systems such as power distribution control.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In

Advances in Neural Information Processing Systems, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martin Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sye57xStvB.

Jonathan Balloch, Zhiyu Lin, Mustafa Hussain, Aarun Srinivas, Xiangyu Peng, Julia Kim, and Mark Riedl. Novgrid: A flexible grid world for evaluating agent response to novelty. In *In Proceedings of AAAI Symposium, Designing Artificial Intelligence for Open Worlds*, 2022.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Terrance Boult, Przemyslaw Grabowicz, Derek Prijatelj, Roni Stern, Lawrence Holder, Joshua Alspector, Mohsen M Jafarzadeh, Toqueer Ahmad, Akshay Dhamija, Chunchun Li, et al. Towards a unifying framework for formal theories of novelty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 15047–15052, 2021.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2018a.

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

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023.

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

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

Felipe Leno Da Silva and Anna Helena Reali Costa. A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, 64:645–703, 2019.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, Sep 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05961-4. URL https://doi.org/10.1007/s10994-021-05961-4.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

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

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

Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021. doi: 10.1177/0278364920987859. URL https://doi.org/10.1177/0278364920987859.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

George Konidaris, Ilya Scheidwasser, and Andrew Barto. Transfer in reinforcement learning via shared features. *Journal of Machine Learning Research*, 13(45):1333–1371, 2012. URL http://jmlr.org/papers/v13/konidaris12a.html.

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

Pat Langley. Open-world learning for radically autonomous agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13539–13543, 2020.

Pat Langley. Agents of exploration and discovery. *Ai Magazine*, 42(4):72–82, 2022.

Alessandro Lazaric. *Transfer in Reinforcement Learning: A Framework and a Survey*, pp. 143–173. Springer Berlin Heidelberg, 2012.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Faizan Muhammad, Vasanth Sarathy, Gyan Tatiya, Shivam Goel, Saurav Gyawali, Mateo Guaman, Jivko Sinapov, and Matthias Scheutz. A novelty-centric agent architecture for changing worlds. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 925–933, 2021.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 2778–2787. JMLR.org, 2017.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByBAl2eAZ.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *International Conference on Learning Representations*, 2019.

Denis Réale, Simon M. Reader, Daniel Sol, Peter T. McDougall, and Niels J. Dingemanse. Integrating animal temperament within ecology and evolution. *Biological Reviews*, 82(2):291–318, 2007. doi: https://doi.org/10.1111/j.1469-185X.2007.00010.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-185X.2007.00010.x.

Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pp. 1458–1463, 1991a.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers, 1991b.

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

Daniel L Silver, Qiang Yang, and Lianghao Li. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*, 2013.

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

Adrien Ali Taiga, William Fedus, Marlos C. Machado, Aaron Courville, and Marc G. Bellemare. On bonus based exploration methods in the arcade learning environment. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJewlyStDr.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Jianye Hao, Zhaopeng Meng, Peng Liu, and Zhen Wang. Exploration in deep reinforcement learning: a comprehensive survey. *arXiv preprint arXiv:2109.06668*, 2021.

Xingrui Yu, Yueming Lyu, and Ivor Tsang. Intrinsic reward driven imitation learning via generative model. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10925–10935. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/yu20d.html.

Mingqi Yuan, Bo Li, Xin Jin, and Wenjun Zeng. Rewarding episodic visitation discrepancy for exploration in reinforcement learning. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022a.

Mingqi Yuan, Man-On Pun, and Dong Wang. Rényi state entropy maximization for exploration acceleration in reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 2022b.

Yusen Zhan and Mattew E Taylor. Online transfer learning in reinforcement learning domains. In *2015 AAAI Fall Symposium Series*, 2015.

Zidong Zhang, Dongxia Zhang, and Robert C Qiu. Deep reinforcement learning for power system applications: An overview. *CSEE Journal of Power and Energy Systems*, 6(1):213–225, 2019.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

A Algorithm Descriptions

RND: Random Network Distillation is an exploration algorithm that uses the error of a randomly generated prediction problem as an intrinsic reward for the agent. The prediction problem is set up with two neural networks: a randomly initialized fixed target network and a predictor network that is attempting to approximate the target network. Both networks take an observation and output a k-dimensional latent vector. The predictor network is trained on observations collected from the agent using gradient descent to minimize the MSE between the outputs of the two neural networks. This MSE loss is used as the intrinsic reward, which will be higher when the predictor network and target network have not been trained on an observation enough to learn the latent yet.

REVD: Rewarding Episodic Visitation Discrepancy is an exploration method that uses intrinsic rewards to motivate the agent to maximize the discrepancy between the set of states visited in consecutive episodes. The discrepancy between consecutive episodes is measured by an estimate of the Renyi divergence using samples from the two episodes. The intrinsic reward is calculated by using the term in the divergence estimate that has to do with the current state, incentivising the agent to visit states that will increase the divergence estimate between the current episode and the previous one.

RE3: Random Encoders for Efficient Exploration is an exploration method that sets the intrinsic reward to an estimate of state entropy. To estimate state entropy, the method applies a k-nearest neighbor entropy estimator in a low-dimensional space the observations are mapped to using a randomly initialized fixed convolutional encoder. The encoder does not need to be trained and instead relies on the convolutional structure of the network, making the algorithm computationally efficient.

RIDE: Rewarding Impact-Driven Exploration is an exploration method that uses intrinsic rewards to incentivize the agent to take actions that lead to large changes in a learned state representation. The learned state representation comes from an encoder that allows for learning of both the forward and inverse models (taken from ICM). The learning problems the state representation is used for only incentivizes the encoder to retain features of the environment that are influenceable by the agent's actions. Thus, the intrinsic reward is defined as the difference in said state representation, allowing the agent to experience a diverse set of states.

ICM: Intrinsic Curiosity Module is an exploration method that uses the prediction error of a forward model that acts on state embeddings as the intrinsic reward. The state embeddings are learned by using these embeddings to learn an inverse model to predict the action that takes a state embedding to the state embedding in the next time step. These state embeddings are learned to only contain information relevant to the inverse model, effectively solving the noisy-tv problem. The prediction error of the forward model as an intrinsic reward motivates the agent to explore states that it has a poor estimate of the forward dynamics, which should correlate with states the agent has observed less.

NGU: Never Give Up is an exploration algorithm that constructs an intrinsic reward to strongly discourage revisiting the same state more than once within an episode and discourage visiting states that have been visited many times before. These goals are achieved by an episodic novelty module and a life-long novelty module respectively. These use the embedding networks trained in the same manner as ICM to generate a meaningful lower dimensional state representation. The episodic novelty module uses episodic memory and a k-nearest neighbors pseudo-count method to calculate the intrinsic reward. The life-long novelty module uses the same method as RND. Then these two values are combined using multiplicative modulation for the final intrinsic reward.

NoisyNets: Noisy Networks is an exploration algorithm that applies parametric noise to the weights to introduce stochasticity in the agent's policy. This method adds very little overhead since all it requires is a few extra noise parameters in a few layers of the network. This added stochasticity in the weights propagates to the agent's policy to lead to the agent exploring more unknown states instead of only acting greedily.

GIRL: Generative Intrinsic Reward Learning is an exploration algorithm that motivates the agent to visit areas in which a separate model attempting to model the conditional state distribution performs poorly. The method does this by adding an intrinsic reward of the reconstruction error of each state to the extrinsic reward from the task. The model used to model the state distribution is a conditional VAE conditioned on the previous state and a latent variable.

RISE: Renyi State Entropy Maximization is an exploration algorithm that uses intrinsic rewards to maximize the estimate of intra episode Renyi state entropy. This estimate is calculated on latent embeddings of the states within an episode, where the latents are taken from a VAE trained to reconstruct the states. Further, the algorithm automatically searches the different possibilities for the value of k used in the KNN for the Renyi state entropy estimation that guarantees estimation accuracy. Lastly, RISE uses the distance between each state and its k-nearest neighbors as an estimate for entropy and sets the intrinsic reward to this value. The goal of this reward is to motivate the agent to visit a diverse set of states that increases the entropy of the agent's state visitations. This method is computationally efficient and does not require any additional memory or networks to backpropagate through.

DIAYN: Diversity Is All You Need is an exploration pre-training method that learns a skill-conditioned policy with the goal to produce diverse skills. This is done by setting the reward to something correlated with the performance of a discriminator model that attempts to predict the skill by using the current state as input. Each episode a new skill is sampled for the policy to use, and the discriminator must attempt to predict the skill. Theoretically, this should lead to the policy attempting to make the job of the discriminator as easy as possible by creating diverse skills. Note that in the original paper this reward and skill-conditioned policy was used before any task reward was introduced. Then, these diverse skills were used to learn a task. However, in our work, we adapt DIAYN to be an online algorithm where this reward is trained simultaneously with the task reward. This motivates the agent to both solve the task while keeping the discriminator's job easy by ensuring different skills cover different areas of the state space. This online adaptation of DIAYN works as a traditional exploration algorithm by motivating the agent to take diverse paths throughout training by sampling different diverse skills to use each episode.

A.1 A note on "online" DIAYN

The effectiveness of explicit diversity and stochasticity methods is consistent throughout our results; however, this does not mean that adding diversity or stochasticity to any algorithm in any way will guarantee improvement to that algorithm's efficiency in novelty adaptation. The fundamental design of an algorithm to succeed in a specific RL problem, such as online task transfer, is as important as the selection of exploration principle and instantiation. For example, online DIAYN has average efficiency in both pre and post-novelty for all tasks we tested it on. However, based on the fact that it blends stochasticity with diverse skills could be interpreted to mean that it ought to have performed better post-novelty. In reality, DIAYN's absence of better performance is more likely due to its implementation; as an algorithm originally designed for reward-free pretraining, naive conversion to an online algorithm, while consistent with the original work and able to learn, is a handicap that cannot be solely compensated for by the potential of its exploration approach. A more transfer-appropriate version of DIAYN—as with all of these algorithms—can be designed from scratch and would likely outperform even the best exploration method investigated here. However, this level of algorithmic design ought to be carefully done with the learning problem in mind and is beyond the scope of this work.

B Exploration Characteristics: Algorithmic Instantiation

We do not report many interesting findings on algorithmic instantiation, partially because our results show that in general algorithmic instantiation does not have an outsized impact on the final results. While NoisyNets with an update function instantiation is consistently high performing in different transfer problems, so is RE3 using an intrinsic reward. Moreover, considering a within-group evaluation of all of the intrinsic reward algorithms, we can see that there is a very high variance over average performance across all metrics; ICM consistently performing poorly, RE3 and REVD consistently performing well, and many of the others performing inconsistently with respect to one another. Maybe most critically, however, we do not think it wise to generalize over conclusions about algorithmic instantiation from this work because of all of the characteristic categories, algorithmic instantiation is the most unbalanced. The vast majority of the algorithms evaluated in this paper are intrinsic reward, while only one, NoisyNets, has a modified update function, and even DIAYN, while altering the environment sampling process by a policy conditioned on a random skill vector, still uses an intrinsic reward as well. This imbalance is accidental, but not unexpected; the vast majority of modern exploration algorithm that generalize to different problems like we used here use intrinsic reward. An important direction of future work will be to construct fair means of comparison with offline algorithms and algorithms only suited for continuous control or discrete control so that more methods like ϵ-greedy (Sutton & Barto, 2018), maximum entropy RL (Hazan et al., 2019; Haarnoja et al., 2018), and replay methods like hindsight experience replay Andrychowicz et al. (2017) can also be compared.

C Additional Analysis

We also examined the shortcut LavaProof novelty as compared to the other novelties, and we see some interesting behavior very specific to the notion of a shortcut. As identified in prior work, shortcuts can be notoriously hard exploration problems for transfer learning because the novelty is injected and the learner's prior optimum is undisturbed. As we have noted, if we used exploration decay in our algorithm implementations, as is common in single-task RL, there is a chance most or even all of the algorithms in this study would ignore the new shortcut and continue with the sub-optimal solution. Even without exploration decay, NGU, GIRL, and ICM all fail to consistently identify the shortcut over the safe lava in spite of learning how to safely navigate around it. Atypically, NoisyNets also performed poorly and was unable to consistently find the novelty. Of those that performed well, in addition to RE3, DIAYN and RIDE performed unusually well. These observations together serve as strong evidence that the main difference in characteristic importance for shortcuts is an even stronger emphasis on the importance of explicit diversity. For a shortcut, the critical steps are to (1) identify that a shortcut exists, and (2) consider it worth exploring. Although intuitively the stochastic nature of NoisyNets may thrive at shortcut identification, it is less likely that a time-independent method like NoisyNets would be able to value exploring something just because it was novel. In this way, the lack of temporal locality in NoisyNets overcomes its potential for exploring the novelty. Interestingly, the reverse happens for DIAYN. DIAYN's core motivation is to learn separable distinguishable policy skills, which for a single task learning problem becomes progressively harder as the policy converges. When a shortcut is identified, there is a novel opportunity for DIAYN to suddenly learn more diverse separable skills. As a result, the DIAYN's specific implementation of explicit diversity is able to overcome its time-independent exploration nature.

D Additional Results

Exploration Algorithm	Convergence Efficiency ↓				
	DoorKeyChange (10^6)	LavaNotSafe (10^5)	LavaProof (10^6)	CrossingBarrier (10^5)	ThighIncrease (10^6)
None (PPO)	2.56 ± 0.584	0.707 ± 0.35	1.7 ± 0.683	5.43 ± 1.69	7.99 ± 1.09
NoisyNets	2.45 ± 0.908	1.02 ± 0.911	1.31 ± 1.14	4.92 ± 2.13	7.17 ± 1.72
ICM	2.12 ± 0.595	0.604 ± 0.0966	1.8 ± 1.46	4.66 ± 1.14	7.34 ± 1.02
DIAYN	2.19 ± 0.808	0.707 ± 0.265	3.44 ± 1.57	5.47 ± 2.37	**6.87 ± 2.41**
RND	2.41 ± 0.956	0.635 ± 0.0893	0.976 ± 0.803	5.11 ± 0.95	7.5 ± 2.04
NGU	2.14 ± 0.289	0.768 ± 0.291	2.34 ± 3.38	5.43 ± 2.03	7.72 ± 1.52
RIDE	2.39 ± 0.975	**0.563 ± 0.0687**	**0.73 ± 0.293**	5.65 ± 2.11	8.24 ± 1.24
GIRL	2.4 ± 0.855	0.676 ± 0.173	2.43 ± 1.69	4.63 ± 0.979	7.61 ± 1.99
RE3	2.14 ± 0.616	0.604 ± 0.107	1.86 ± 0.669	5.42 ± 1.37	7.78 ± 0.642
RISE	2.32 ± 0.764	0.614 ± 0.145	3.14 ± 1.89	**4.29 ± 0.788**	8.55 ± 0.441
REVD	**2.12 ± 0.891**	0.635 ± 0.188	1.72 ± 1.66	4.8 ± 1.12	8.73 ± 0.934

Table 4: This table shows the convergence efficiency on the pre-novelty task. It is computed by calculating the number of steps from the start of training until convergence on the first task. Thus, lower numbers are better here. Only runs that converged on the first task are taken into account for this metric.

Exploration Algorithm	Adaptive Freq ↑				
	DoorKeyChange	LavaNotSafe (10^{-1})	LavaProof	CrossingBarrier	ThighIncrease
None (PPO)	**1.0 ± 0.0**	6.0 ± 4.9	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
NoisyNets	0.889 ± 0.314	**8.0 ± 4.0**	0.714 ± 0.452	**1.0 ± 0.0**	**1.0 ± 0.0**
ICM	0.889 ± 0.314	3.0 ± 4.58	0.875 ± 0.331	**1.0 ± 0.0**	**1.0 ± 0.0**
DIAYN	**1.0 ± 0.0**	3.0 ± 4.58	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
RND	**1.0 ± 0.0**	3.0 ± 4.58	0.714 ± 0.452	**1.0 ± 0.0**	**1.0 ± 0.0**
NGU	**1.0 ± 0.0**	4.0 ± 4.9	**1.0 ± 0.0**	0.9 ± 0.3	**1.0 ± 0.0**
RIDE	**1.0 ± 0.0**	6.0 ± 4.9	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
GIRL	**1.0 ± 0.0**	2.0 ± 4.0	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
RE3	**1.0 ± 0.0**	2.0 ± 4.0	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
RISE	0.857 ± 0.35	3.0 ± 4.58	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**
REVD	**1.0 ± 0.0**	5.0 ± 5.0	**1.0 ± 0.0**	**1.0 ± 0.0**	**1.0 ± 0.0**

Table 5: This is the frequency that the agent converges on the second task using this exploration algorithm conditioned on the fast it converged on the first task. Higher numbers are better.

E Implementation Details

E.1 Hyperparameters

We sweeped through the hyperparameter configurations for each exploration algorithm using Bayesian hyperparameter optimization. We ran a minimum of 10 hyperparameter configurations (using more for the algorithms with many parameters), each with six runs (three seeds on MiniGrid-DoorKey-8x8-v0 and three seeds on MiniGrid-SimpleCrossingS9N2-v0), for each algorithm. Each successive configuration was calculated using the weights and biases Bayesian sweep method within reasonable preset range around parameters pulled from prior work. The metric optimized for to minimize the average (over the 6 runs) number of steps needed for the StopTrainingOnRewardThreshold callback from stable-baselines3 to stop the run with a reward threshold set to 0.35 (capped at 3M steps). Once the sweeps were finished we chose reasonable hyperparameters that followed the trends of the other runs in the sweep to ensure the chosen parameter configuration was not just an outlier.

Here is a table consisting of the ranges of hyperparameters we sweeped through and our final chosen value for them based on the (limited) number of runs we used. The distribution type column refers to the distribution parameter provided to the wandb sweep agent. For specifics about what each parameter does see the individual papers or the implementations in our codebase. Note that latent_dim, batch_size, and learning_rate parameters refer to networks trained specifically for exploration and have nothing to do with the parameters used for policy training.

Algorithm	Parameter Name	Distribution Type	Range	Final Value
PPO	learning_rate	q_uniform	[0.0003, 0.0008]	0.00075
RE3	beta	q_log_uniform_values	[0.00001, 0.1]	0.01
	latent_dim	categorical	[16, 32, 64, 128, 256]	64
RIDE	beta	q_log_uniform_values	[0.00001, 0.1]	0.001
	latent_dim	categorical	[16, 32, 64, 128, 256]	128
RISE	beta	q_log_uniform_values	[0.00001, 0.1]	0.002
	latent_dim	categorical	[16, 32, 64, 128, 256]	64
RND	beta	q_log_uniform_values	[0.00001, 0.1]	0.002
	learning_rate	q_log_uniform_values	[0.0001, 0.01]	0.0003
	batch_size	categorical	[16, 32, 64]	64
	latent_dim	categorical	[16, 32, 64, 128, 256]	128
Noisy Nets	num_noisy_layers	categorical	[1, 2, 3]	2
NGU	beta	q_log_uniform_values	[0.0001, 0.5]	0.0005
	learning_rate	q_log_uniform_values	[0.0001, 0.01]	0.0006
	batch_size	categorical	[16, 32, 64]	64
	latent_dim	categorical	[16, 32, 64, 128, 256]	128
ICM	beta	q_log_uniform_values	[0.00001, 0.1]	0.0003
	learning_rate	q_log_uniform_values	[0.0001, 0.01]	0.0003
	batch_size	categorical	[16, 32, 64]	64
GIRL	beta	q_log_uniform_values	[0.00001, 0.1]	0.0005
	learning_rate	q_log_uniform_values	[0.0001, 0.01]	0.002
	lambda	q_log_uniform_values	[0.001, 0.1]	0.05
	latent_dim	categorical	[32, 64, 128]	64
REVD	beta	q_log_uniform_values	[0.00001, 0.1]	0.00005
	latent_dim	categorical	[16, 32, 64, 128, 256]	64
RIDE	beta	q_log_uniform_values	[0.00001, 0.1]	0.001
	latent_dim	categorical	[16, 32, 64, 128, 256]	128
RISE	beta	q_log_uniform_values	[0.00001, 0.1]	0.002
	latent_dim	categorical	[16, 32, 64, 128, 256]	64

Table 6: Hyperparameter Sweeps for Exploration Algorithms.

For the continuous control task (Walker), we ran a targeted sweep on CartPole, mainly tuning parameters that were important to our results such as beta and other exploration algorithm specific parameters. We used prior work, results from our MiniGrid sweep, and other heuristics to estimate the ranges to sweep for different parameters. The main parameters that changed relative to the table above were the beta's for each algorithm as the reward scale is very different in walker as opposed to any MiniGrid tasks.

E.2 Experimental Setup

For a valid comparison, all the experiments were run using PPO with the same PPO hyperparameters (listed below). Further, the experiments use the default MLP policy network shapes from the stable-baselines3 PPO class for the experiments and any hyperparameters not specified below were left as default.

Parameter	Value
learning_rate	0.00075
n_steps	2048
batch_size	256
n_epochs	4
gamma	0.99
gae_lambda	0.95
clip_range	0.2
ent_coef	0.01
vf_coef	0.5
max_grad_norm	0.5

Table 7: PPO Configuration

Each experiment on MiniGrid used 10 seeds with 5 parallel environments each to ensure reliable results, logging all results to wandb for future aggregation and analysis.

Each experiment on Walker used 5 seeds with 10 parallel environments.

For each of the environments, we ran the experiments with a number of steps that led to a high convergence rate with the implemented algorithms so fair comparisons between algorithms could be used on the task two results.

Environment Name	Pre Novelty Steps	Post Novelty Steps	MiniGrid Size
door_key_change	5M	3M	8x8
simple_to_lava_crossing	2M	3M	9x9
lava_maze_safe_to_hurt	500,000	5M	8x8
lava_maze_hurt_to_safe	5M	2M	8x8
walker_thigh_length	10M	10M	N/A

Table 8: Environment Details

We used a few observation wrappers on the environments in the experiment to set the observation space to be the flattened observed image (to work with simple MLP policies).