# DEIR: Efficient and Robust Exploration through Discriminative-Model-Based Episodic Intrinsic Rewards

**Shanchuan Wan**[1] , **Yujin Tang**[2] , **Yingtao Tian**[2] and **Tomoyuki Kaneko**[1]
[1]The University of Tokyo
[2]Google Research, Brain Team
swan@game.c.u-tokyo.ac.jp, {yujintang, alantian}@google.com, kaneko@graco.c.u-tokyo.ac.jp

## Abstract

Exploration is a fundamental aspect of reinforcement learning (RL), and its effectiveness is a deciding factor in the performance of RL algorithms, especially when facing sparse extrinsic rewards. Recent studies have shown the effectiveness of encouraging exploration with intrinsic rewards estimated from novelties in observations. However, there is a gap between the novelty of an observation and an exploration, as both the stochasticity in the environment and the agent's behavior may affect the observation. To evaluate exploratory behaviors accurately, we propose **DEIR**, a novel method in which we theoretically derive an intrinsic reward with a conditional mutual information term that principally scales with the novelty contributed by agent explorations, and then implement the reward with a discriminative forward model. Extensive experiments on both standard and advanced exploration tasks in MiniGrid show that DEIR quickly learns a better policy than the baselines. Our evaluations on ProcGen demonstrate both the generalization capability and the general applicability of our intrinsic reward.

## 1 Introduction

Exploration is an important aspect of reinforcement learning (RL), as suggested by the famous exploration-exploitation trade-off [Sutton and Barto, 2018] wherein an agent that only exploits with the current policy would be stuck and fail to improve its policy anymore due to the lack of novel experiences. Effective exploration is non-trivial, especially in tasks where environmental rewards are sparse. Relying on unstructured exploration (e.g., $\epsilon$-greedy or randomized probability matching [Sutton and Barto, 2018; Scott, 2010]) requires an exponential number of samples and is unlikely to achieve a satisfactory level of exploration. Manually designing dense rewards with domain knowledge has exhibited promising results in several areas where RL has significantly progressed, such as game-playing and robotics [Mnih *et al.*, 2015; Baker *et al.*, 2019; Hafner *et al.*, 2020]. However, given the huge amount of knowledge and effort required, designing such dense rewards is only feasible in a handful of tasks, and effective exploration thus remains a challenge.

To tackle this issue, several works have proposed guiding the exploration with intrinsic rewards or rewards that are internal to agents, including ICM [Pathak *et al.*, 2017], RND [Burda *et al.*, 2019], NGU [Badia *et al.*, 2020], and NovelD [Zhang *et al.*, 2021]. In these methods, the intrinsic reward is devised to encourage visiting states that are likely

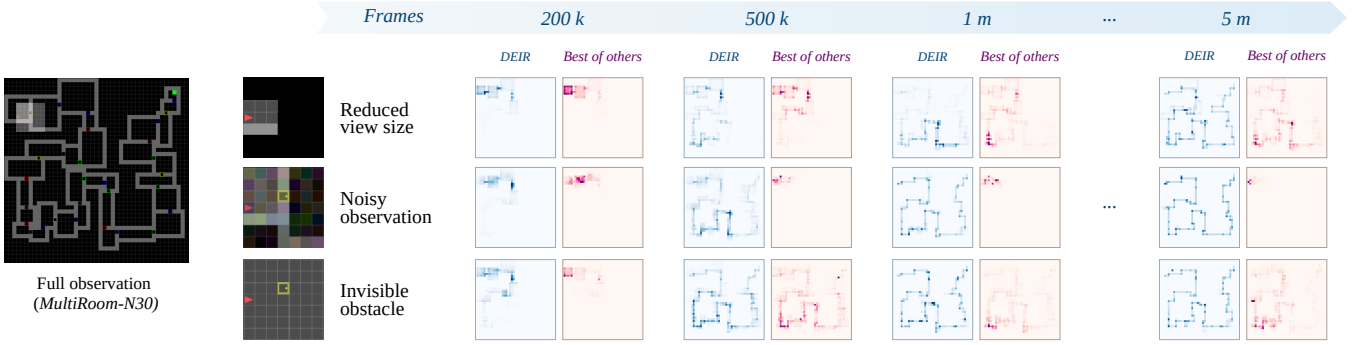


Figure 1: Enlarged *MultiRoom* environment (leftmost figure) from MiniGrid with 30 cascaded rooms as a representative of environments with sparse extrinsic rewards, where the agent (upper-left red dot) is tasked with finding the optimal path to the goal (upper-right green dot). We create advanced variants (three rows on the right) with extra difficulties (reduced view sizes, noisy observations, and invisible obstacles). In these challenging tasks, existing methods (ICM, RND, NGU, and NovelD) either fail to find an optimal path or require a prohibitive number of episodes to do so. In contrast, our proposed DEIR leverages a conditional mutual information-based intrinsic reward and a contrastive learning-inspired model, which is capable of diverse exploration and delivering significantly better performances (see Section 4.2).

to be more novel, where novelty is defined as either the distance between the current and past observations or the difference between model predictions and realities. Although these works have shown encouraging empirical results leading to better exploration efficiency, the relationship between the observed novelty and the agent's actions has not yet been explicitly decoupled. In other words, effectively handling trivial novelties remains unaddressed, as they are rooted in the stochasticity in the environment's dynamics and have little to do with the agent's exploration capabilities (e.g., the "noisy TV" problem [Pathak *et al.*, 2017]).

This paper bridges this gap with **DEIR** (**D**iscriminative-model-based **E**pisodic **I**ntrinsic **R**eward), a novel intrinsic reward design that considers not only the observed novelty but also the effective contribution brought by the agent. In DEIR, we theoretically derive an intrinsic reward by scaling the novelty metric with a conditional mutual information term between the observation distances and the agent's actions. We make the intrinsic reward tractable by using a simple formula serving as the lower bound of our primary objective. DEIR is thus designed to distinguish between the contributions to novelty caused by state transitions and by the agent's policy. For computing the proposed reward, we devise a discriminative forward model that jointly learns the environment's dynamics and the discrimination of genuine and fake trajectories. We evaluated DEIR in both standard and advanced MiniGrid [Chevalier-Boisvert *et al.*, 2018] tasks (grid-world exploration games with no extrinsic reward until reaching the goal) and found that it outperforms existing methods on both (see Figure 1). To examine DEIR's generalization capability in tasks with higher dimensional observations, we also conducted experiments in ProcGen [Cobbe *et al.*, 2019; Cobbe *et al.*, 2020] (video games with procedurally generated levels that require planning, manipulation, or exploration) and found that it delivers a state-of-the-art (SOTA) performance in all selected tasks. Finally, we performed an in-depth analysis of our method through additional experiments to clarify the effectiveness of DEIR and its components.

Our contributions can be summarized as follows. (1) Our method, theoretically grounded, effectively decouples the stochasticity in the environment and the novelty gained by the agent's exploration. (2) Our method empirically outperforms others in this line of work by a large margin, especially in advanced MiniGrid tasks, but not limited thereto. It also has potential applications for a variety of other tasks. (3) Our method is easy to implement and use, and provides state representations useful for potential downstream objectives.

## 2 Related Work

Exploration through intrinsic rewards is widely studied in RL literature, with many works falling into one of two categories: *Prediction error-driven methods* that are motivated by the differences ("surprises") between predictions and realities, and *Novelty-driven methods* that seek novel agent observations.

**Prediction error-driven methods.** It is typical for prediction error-driven methods to learn a model of the environment's dynamics and use it to make predictions for future states. Large deviations between the predictions and the realities suggest regimes where the model is insufficiently learned. Intrinsic rewards, positively correlated to prediction errors, encourage the agent to explore more in those states. One of the representative works in this category is ICM [Pathak *et al.*, 2017], which jointly trains both the forward and the inverse transition models to capture the environment's dynamics better, but only uses the forward model's prediction errors to generate intrinsic rewards for the agent. Prediction error-driven methods require approximating the environment's dynamics with neural networks, which is especially difficult in high-dimensional spaces. Still, as demonstrated by ICM, training with auxiliary tasks seems to be worth the effort.

**Novelty-driven methods.** Recent novelty-driven methods usually define a distance metric between observations, and then formulate this distance as intrinsic rewards to encourage more exploration in RL roll-outs. Early works include count-based methods, which record how many times distinct states are visited and use the count differences as intrinsic rewards [Bellemare *et al.*, 2016; Ostrovski *et al.*, 2017; Tang *et al.*, 2017]. However, due to their simplicity, count-based methods have difficulty in tasks featuring continuous states or high-dimensional observations. To overcome this problem, neural networks have been introduced to encode the states and observations. For example, RND [Burda *et al.*, 2019] defines the distance as the difference between the outputs of a parameter-fixed target neural network and a randomly initialized neural network. In this approach, the former network is used to be distilled into the latter one, effectively "evolving" a distance metric that adjusts dynamically with the agent's experience. In NovelD [Zhang *et al.*, 2021], one of the latest works, RND is applied to evaluate the distances between pairs of observations, in which the boundary between explored and unexplored regions is defined as where the distance is larger than a predefined threshold and large intrinsic rewards are provided when the agent crosses the boundaries. In this way, NovelD encourages the agent to explore in a manner similar to breadth-first search and has already demonstrated a state-of-the-art (SOTA) performance on many MiniGrid tasks. Never-Give-Up (NGU) [Badia *et al.*, 2020] introduces the inverse model from ICM in its episodic intrinsic reward generation module. Their final intrinsic reward is based on the Euclidean distances of the K-nearest embeddings of the recently visited states. In practice, measuring novelty often requires analysis of the distribution of an environment's states. Even so, "noisy TV"-like problems can still occur, where novelty in the observation space is primarily due to the stochasticity in an environment's dynamics. This prevents the agent from achieving meaningful explorations.

Our method incorporates the advantages of the two categories: while we explicitly encourage our agent to seek novel observations, we also rely on a discriminative model to construct a conditional mutual information term that scales novelty in the observation space, incorporating the model-based prediction task from prediction error-driven methods. Unlike conventional novelty-driven methods, our conditional mutual information scaling term effectively eliminates the novelties rooted in the environment's stochasticity other than those brought by the agent's explorations. It has granted us better

performances (see Figure 1). To illustrate the difference from existing prediction error-driven methods, our model learns both the environment's dynamics and the capability to tell the genuine and fake trajectories apart. Consistent with the results reported in contrastive learning studies [Laskin *et al.*, 2020; Agarwal *et al.*, 2021], the discriminative nature of our model encodes the observations in a space closely related to the underlying tasks and thus enables us to measure the distances between observations more accurately.

# 3 Proposed Method

## 3.1 Background and Notations

A Markov decision process (MDP) [Sutton and Barto, 2018] can be defined as a tuple $(S, A, r, f, P_0, \gamma)$, where $S$ is the state space, $A$ is the action space, $r(s_t, a_t, s_{t+1})$ is the reward function, $f(s_t, a_t)$ is the environment's transition function, $P_0$ is the distribution of the initial state $s_0$, and $\gamma \in [0, 1]$ is the reward discount factor. The goal is to optimize a policy $\pi : S \times A \to \mathbb{R}$ so that the expected accumulated reward $\mathbb{E}_{s_0 \sim P_0}[\sum_t \gamma^t r(s_t, a_t, s_{t+1})]$ is maximized. However, in a partially observable MDP (POMDP), $s_t \in S$ is not accessible to the agent. A common practice is to use $\pi(a_t|\tau_t)$ for the policy instead, where $\tau_t = \{o_0, o_1, \cdots, o_t\}$ is an approximation of $s_t$. Many works implement this using recurrent neural networks (RNN) [Rumelhart *et al.*, 1986] to best utilize available historical information.

We adopt proximal policy optimization (PPO) [Schulman *et al.*, 2017] to learn the agent's policy. With PPO as a basis, we propose an episodic intrinsic reward that helps decouple the novelties introduced by the agent from those by the environment. We also introduce a discriminative forward model to learn better state representations in partially observable environments. Following popular studies, our reward function is designed to be the weighted sum of extrinsic (those from the environment) and intrinsic (those from curiosity) rewards: $r(s_t, a_t, s_{t+1}) = r_t^{\mathrm{E}} + \beta \cdot r_t^{\mathrm{I}}$, where $r_t^{\mathrm{E}}$ and $r_t^{\mathrm{I}}$ (both functions of $(s_t, a_t, s_{t+1})$) are respectively the extrinsic and intrinsic rewards at time step $t$, and $\beta$ is a hyperparameter.

## 3.2 Episodic Intrinsic Reward

### Scaling the Novelty

For a pair of states $(s_t, s_i), \forall i \in [0, t)$ and the action $a_t$, we want a novel state (valuable for exploration) to have a large distance between observations $(o_{t+1}, o_i)$ while that distance is closely related to the action $a_t$. Intuitively, it is crucial to distinguish novelty rooted in the environment's stochasticity from novelty brought by an exploratory action of the agent. This leads to our primary objective, which is to maximize

$$J = \mathrm{dist}(o_{t+1}, o_i) \cdot I\left(\mathrm{dist}(o_{t+1}, o_i); a_t | s_t, s_i\right) \qquad (1)$$

as a product of the distance $\mathrm{dist}(o_{t+1}, o_i)$ (denoted as $D_{t+1,i}$) between observations and conditional mutual information

$$I(D_{t+1,i}; a_t | s_t, s_i) = \mathop{\mathbb{E}}_{s_t, s_i, a_t} [\mathrm{D_{KL}}(p(D_{t+1,i}|s_t, s_i, a_t) \| p(D_{t+1,i}|s_t, s_i))].$$

With the Bretagnolle–Huber inequality [Bretagnolle and Huber, 1978], we have

$$\mathrm{D_{KL}}(P\|Q) \geq -\log(1 - \mathrm{d}_{\mathrm{TV}}^2(P, Q)),$$

where $P(x) = p(x|s_t, s_i, a_t)$ and $Q(x) = p(x|s_t, s_i)$ are defined for simplicity, and $\mathrm{d_{TV}}(P, Q) = \frac{1}{2}\|P - Q\|_1$ is the total variation between $P$ and $Q$.

Note that, in deterministic environments (including partially observable cases), (a) $P$ is a unit impulse function that has the only non-zero value at $D_{t+1,i}$, and (b) we can naturally assume the $D_{t+1,i}|s_t, s_i \sim \mathrm{Exp}(\lambda = 1/\mathrm{dist}(s_t, s_i))$ to match the distance between observations and that of their underlying states. Thus, we devise a simplified surrogate function for the mutual information, as

$$\mathrm{D_{KL}}(P\|Q) \geq \log(\mathrm{dist}(s_t, s_i)) + \frac{\mathrm{dist}(o_{t+1}, o_i)}{\mathrm{dist}(s_t, s_i)} + \mathrm{const.}$$

Substituting it back to the objective, we obtain

$$J \geq \mathrm{dist}(o_{t+1}, o_i) \cdot \mathop{\mathbb{E}}_{s_t, s_i, a_t} \left( \log(\mathrm{dist}(s_t, s_i)) + \frac{\mathrm{dist}(o_{t+1}, o_i)}{\mathrm{dist}(s_t, s_i)} \right). \qquad (2)$$

To make it tractable, we simplify the right-hand side to

$$J \geq \min_i \frac{\mathrm{dist}^2(o_{t+1}, o_i)}{\mathrm{dist}(s_t, s_i)}, \qquad (3)$$

which is a lower bound for the original objective. It is simpler and empirically performs as well as or even better than Equation 2. We speculate that improving the minimum value is crucial to improving the expectation value. A detailed derivation is provided in Appendix A.1.

### Intrinsic Reward Design

To maximize our objective through the lower bound in Equation 3, we propose our intrinsic reward in an episodic manner:

$$r_t^{\mathrm{I}} = \min_{\forall i \in [0,t)} \left\{ \frac{\mathrm{dist}^2(e_i^{\mathrm{OBS}}, e_{t+1}^{\mathrm{OBS}})}{\mathrm{dist}(e_i^{\mathrm{TRAJ}}, e_t^{\mathrm{TRAJ}}) + \epsilon} \right\}, \qquad (4)$$

where $e_t^{\mathrm{OBS}}$ is the embedding of observation $o_t$, $e_t^{\mathrm{TRAJ}}$ is the embedding of trajectory $\tau_t$ at time step $t$ in an episode, dist is the Euclidean distance between two embeddings, and $\epsilon$ is a small constant ($10^{-6}$ in our experiments) for numeric stability. Note that the complete information regarding $s_t$ is inaccessible in POMDPs, so $e_t^{\mathrm{TRAJ}}$ is commonly used as a proxy of $s_t$. All these constitute a metric space of observations where distance is defined. Both $e_t^{\mathrm{OBS}}$ and $e_t^{\mathrm{TRAJ}}$ are computed by a discriminative forward model, as detailed in Section 3.3.

Our intrinsic rewards are episodic, as they are created from the observations seen in a single episode. Arguably, episodic rewards are generally compatible with lifelong rewards and can be used jointly. Still, in this work, we focus on the former for simplicity and leave such a combination for future studies.

## 3.3 Learning a Discriminative Model

We train a neural network to extract embeddings $e_t^{\mathrm{OBS}}$ and $e_t^{\mathrm{TRAJ}}$ from observations in high-dimensional spaces. Existing studies have adopted auxiliary tasks in which the forward and inverse models are two representatives to obtain better embeddings suitable for exploration. We propose an improved auxiliary task suitable for exploration in POMDPs inspired by contrastive learning [Laskin *et al.*, 2020]. Concretely, our proposed model learns the environment's dynamics and discriminates the genuine trajectories from the fake. Figure 2 illustrates the architecture of our model.

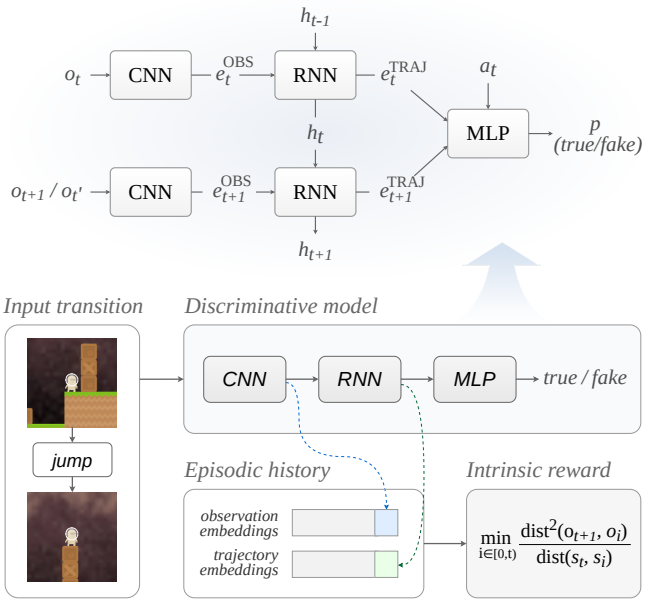


Figure 2: Overview of proposed DEIR. Given the *input transition* of two observations and an action between them, the *discriminative model* predicts whether they are from a truly observed transition. Observation and trajectory embeddings produced by the model are saved in an *episodic history* to compute *intrinsic rewards* for guiding explorations during RL roll-outs.

### Model Definition

Our discriminative model is denoted as $\text{Dsc}(o_t, a_t, o_x) : O \times A \times O' \to [0, 1]$, where $o_t, a_t$ are defined as above and $o_x \in \{o_{t+1}, o_{t'}\}$ is either the next observation $o_{t+1}$ (positive example) or an observation $o_{t'}$ that has been observed recently and is randomly selected at time step $t$ (negative example) for each training sample. In short, the model estimates the likelihood that the input $o_x$ is a positive example.

Specifically, to efficiently retrieve fake examples with decent diversity during RL roll-outs, we maintain recent novel observations with a first-in-first-out queue $\mathbb{Q}$ so that $o_{t'}$ can be randomly selected from all observations saved in $\mathbb{Q}$. This queue is maintained by continuously adding newly retrieved "novel" observations to replace the oldest observations. Here, $o_t$ is considered "novel" only if $r_t^I$ is not less than the running average of all intrinsic rewards. In addition, when sampling fake observations from $\mathbb{Q}$, we always sample twice and keep only ones that differ from the true ones. These measures lead to a very high ratio of valid samples with distinct true and fake observations in training.

### Proposed Architecture

The proposed intrinsic reward is based on the observation embeddings $e_t^{\text{OBS}}$ and the trajectory embeddings $e_t^{\text{TRAJ}}$ generated by the discriminator. From input to output, the discriminator Dsc is formed by a convolutional neural network (CNN) [LeCun *et al.*, 1998], a recurrent neural network (RNN) [Rumelhart *et al.*, 1986], and a multi-layer perceptron (MLP) [Rosenblatt, 1958] output head. We adopt gated recurrent units (GRU) [Cho *et al.*, 2014] in the RNN module to reduce the number of trainable parameters (compared to LSTM [Hochreiter and Schmidhuber, 1997]). As shown in Figure 2, the CNN takes the observations $o_t$ and $o_{t+1}$ as input in parallel and outputs two observation embeddings. These observation embeddings are then fed into the RNN, together with the RNN's previous hidden state $h_{t-1}$. In addition to the updated hidden state $h_t$, the RNN outputs the embeddings of two trajectories starting from the beginning of an episode and ending at time steps $t$ and $t + 1$, respectively. Finally, the two trajectory embeddings with the action $a_t$ are fed into the MLP for predicting the likelihood. RNN hidden states are saved as part of the discriminative model's training samples. We adopt PPO for learning the policy, as it refreshes the experience buffer more frequently than other algorithms. Therefore, we do not apply specialized methods to renew the hidden states within one episode.

### Mini-Batches and Loss Function

During training, each mini-batch consists of two types of samples: half of them positive and half of them negative. Both types are picked from the agent's recent experiences. The discriminator is trained with a binary cross-entropy loss function [Murphy, 2022] as in ordinary classification tasks.

## 4 Experiments

We address the following questions through our experiments:

- Is DEIR effective in standard benchmark tasks and can it maintain decent performance in advanced, more challenging settings?
- Is our design decision in DEIR generally applicable to a variety of tasks, and particularly, can it generalize to tasks with higher dimensional observations?
- How significantly does each technical component in DEIR contribute to the performance?

### 4.1 Experimental Setup

We evaluate DEIR using the following two popular procedurally generated RL benchmarks. (1) **MiniGrid** [Chevalier-Boisvert *et al.*, 2018], which consists of 20 grid-world exploration games featuring different room layouts, interactive objects, and goals. An agent needs to learn a specific sequence of actions to reach a final goal with its limited view size. Valid actions include picking up a key, unlocking a door, unpacking a box, and moving an object. No extrinsic reward is given until the goal. (2) **ProcGen** [Cobbe *et al.*, 2020], which consists of 16 games with $64 \times 64 \times 3$ RGB image inputs, each of which requires a certain level of planning, manipulation, or exploration skill to pass. Each episode is a unique game level with randomly initialized map settings, physical properties, enemy units, and visual objects. The agent needs to learn policies that can be generalized to unseen levels.

Environments and networks are by default initialized with 20 seeds in each MiniGrid experiment and three seeds (from the full distribution of game levels) in each ProcGen experiment. All experimental results are reported with the average episodic return of all runs with standard errors. We performed hyperparameter searches for every method involved in our experiments to ensure they have the best performance possible.

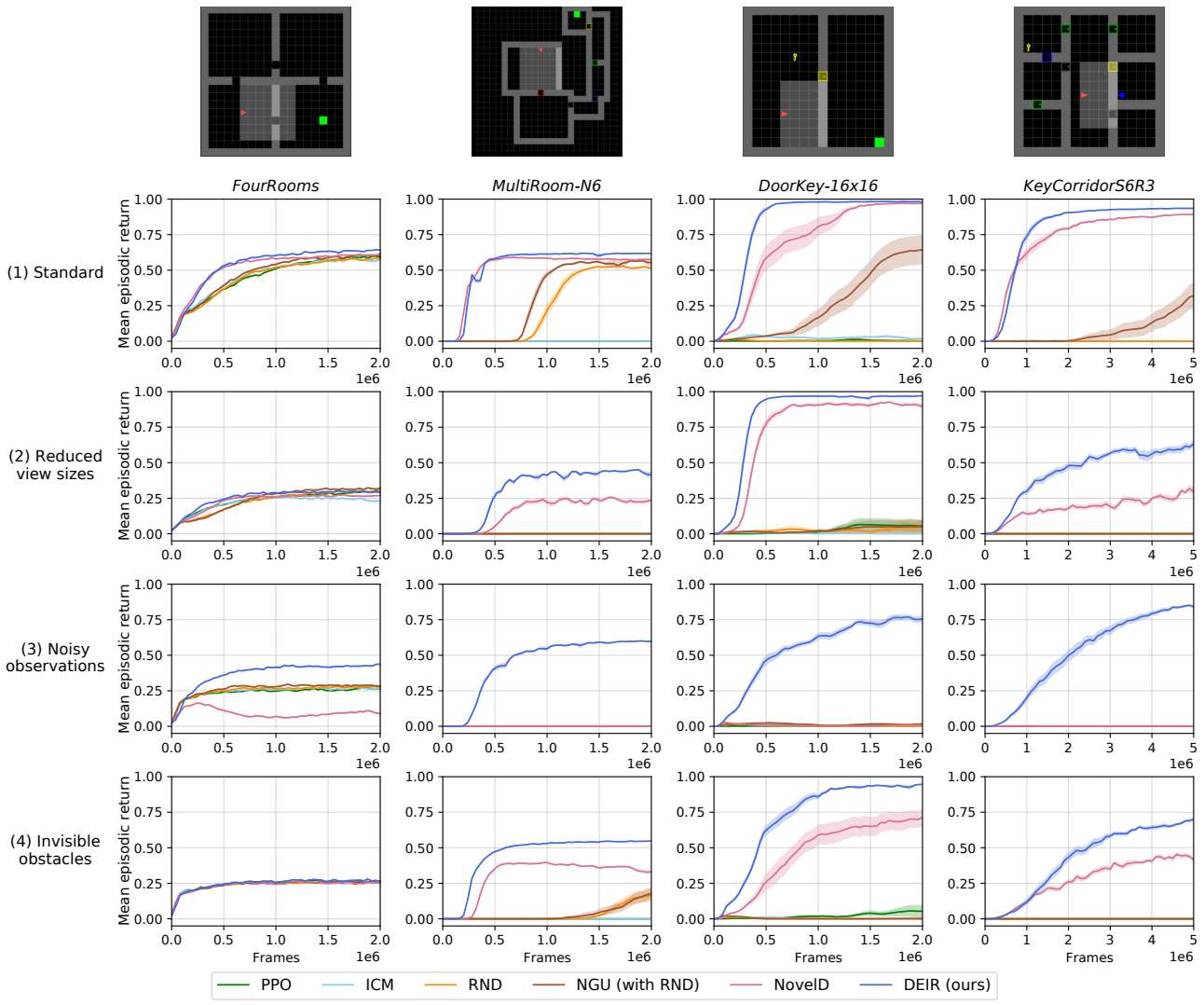


Figure 3: Mean episodic returns in (1) standard and (2)–(4) advanced MiniGrid games. (1) Agent has a fixed $7 \times 7$, unhindered view size. (2) Agent has a reduced view size of $3 \times 3$ grids. (3) Noisy observations, where Gaussian noise ($\mu = 0.0, \sigma = 0.1$) are added element-wise to the observations that are first normalized to $[0, 1]$. (4) Obstacles that are invisible to the agent but still in effect. In all figures, Y axes start at $-0.05$ to show near-zero values.

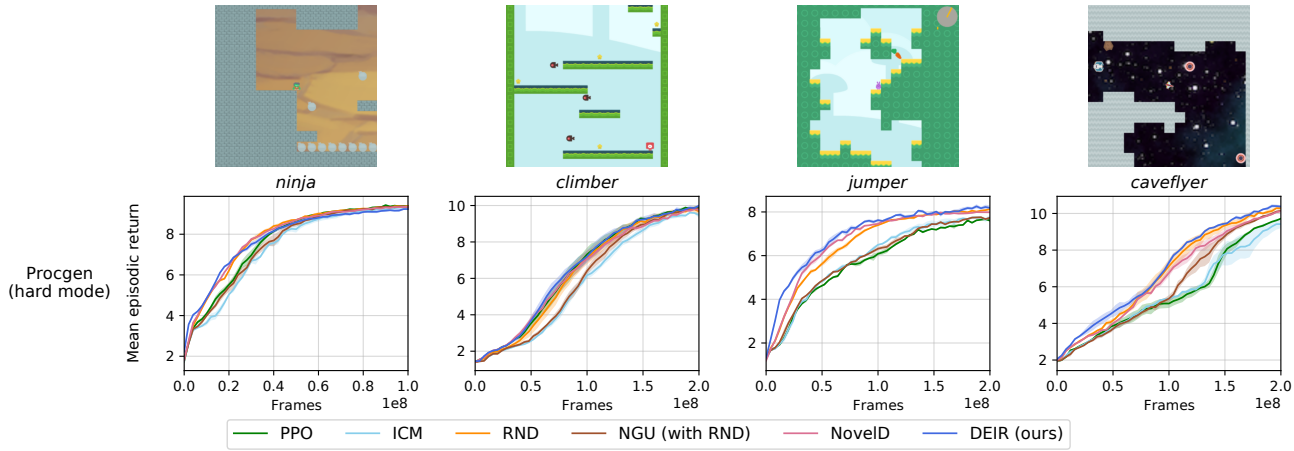


Figure 4: Mean episodic returns in ProcGen (hard mode) games. Episodes are randomly generated from the full distribution of game levels.

We also performed sensitivity analyses on two key hyperparameters of our method, namely, the maximum episode length and the maximum observation queue size, and found that both work well with a wide range of candidate values.

Note that, while our experiments in this paper focus on POMDPs, we find that the performance of DEIR is superior to existing exploration methods in many fully observable tasks as well, which deserves further analysis in future works.

## 4.2 Evaluation Experiments

### Evaluation in Standard MiniGrid

We first evaluated the performance of DEIR in four standard MiniGrid tasks, where the agent's view size is a $7 \times 7$ square and no other constraints are applied. The mean episodic returns of all exploration methods are shown in Figure 3 (first row). In *FourRooms* and *MultiRoom-N6*, which are simple tasks, DEIR and the existing methods all exhibited a decent performance. *DoorKey-16x16* and *KeyCorridorS6R3* are more complex tasks featuring multiple sub-tasks that must be completed in a particular order, requiring efficient exploration. In these complex tasks, DEIR also learned better policies faster than the existing methods.

We also compared the performances of DEIR and NovelD on the most difficult MiniGrid task *ObstructedMaze-Full* (see Figure 5). Its difficulty is due to the highest number of subtasks that need to be completed in the correct order among all standard MiniGrid tasks. So far, it has been solved by only few methods, including NovelD, albeit in an excessive amount of time. To reach the same SOTA performance, DEIR required only around $70\%$ of frames as NovelD, for which we also conducted hyperparameter searches and exceeded its original implementation by requiring fewer training steps.

### Evaluation in Advanced MiniGrid

We further evaluated the robustness of DEIR on 12 MiniGrid tasks with advanced (more challenging) environmental settings, in which the following modifications were made to the standard environments (see examples in Figure 1):

**Reduced view sizes.** The agent's view size is reduced from the default $7 \times 7$ grids to the minimum possible view size of $3 \times 3$ (81.6% reduction in area). Consequently, the agent needs to utilize its observation history effectively.

**Noisy observations.** At each time step, noises are sampled from a Gaussian distribution ($\mu = 0.0, \sigma = 0.1$) in an element-wise manner and added to the observations that are first normalized to $[0, 1]$. With this change, each observation looks novel even if the agent does not explore.

**Invisible obstacles.** Obstacles are invisible to the agent but still in effect; that is, the agent simply perceives them the same way as floors, but cannot step on or pass through them. This requires the agent to have a comprehensive understanding of the environment's dynamics, beyond the superficial observable novelty.

The results in Figure 3 (second to fourth rows) clearly show that DEIR was significantly more robust than the existing methods in all advanced settings. The performance difference was especially large when only noisy observations were presented. Compared to the results in standard tasks, DEIR lost

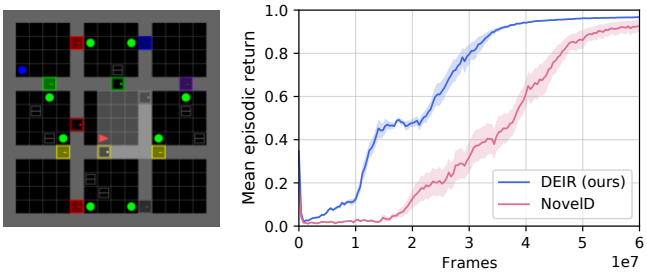


Figure 5: Comparison of episodic returns of DEIR and NovelD in *ObstructedMaze-Full*—the hardest standard task in MiniGrid that has been solved by only few methods, including NovelD. DEIR was able to solve the task here and also learned significantly faster.

at most 25% of its returns by the end of the training, while other methods lost nearly all of their returns in 75% of the noisy-observation tasks. Note that our noisy-observation task is similar to the "noisy-TV" experiments [Burda *et al.*, 2019] but features increased difficulty due to changing more pixels. In ICM [Pathak *et al.*, 2017], up to 40% of the observation image was replaced with white noise. According to published source code, NovelD was originally tested in a scenario where at most one special object in an observation switches its color among six hard-coded colors when triggered by the agent, i.e., most parts of the observation image remain unchanged.

### Generalization Evaluation in ProcGen

We evaluated the generalization capability of DEIR under various reward settings and environmental dynamics using the ProcGen benchmark. Four tasks (on hard mode) were selected on the basis of their partial observability and demand for advanced exploration skills. We used the same CNN structure as in IMPALA [Espeholt *et al.*, 2018] and Cobbe et al.'s work [Cobbe *et al.*, 2020] for the agent's policy and value function, and a widened version of the CNN used in DQN [Mnih *et al.*, 2015] for the dynamics model of each exploration method. The results in Figure 4 show that DEIR performed better than or as well as other exploration methods and could successfully generalize to new game levels generated during training. The results also suggest that the proposed model and intrinsic reward are universally applicable to a variety of tasks, including those with higher-dimensional observations. In addition, we confirmed that the performance of DEIR was consistent with the training results reported in previous studies [Cobbe *et al.*, 2020; Cobbe *et al.*, 2021; Raileanu and Fergus, 2021].

## 4.3 Ablation Studies

To better understand DEIR, we created an advanced *KeyCorridorS6R3* task with all three modifications proposed in Section 4.2 (view size: 3, standard deviation of noise: 0.3, obstacles: invisible), and conducted ablation studies to analyze the importance of (1) the conditional mutual information term and (2) the discriminative model.

### Conditional Mutual Information Scaling

To analyze the importance of the conditional mutual information term proposed in Equation 1, we evaluated the performances of our DEIR agent and a variant without the mutual

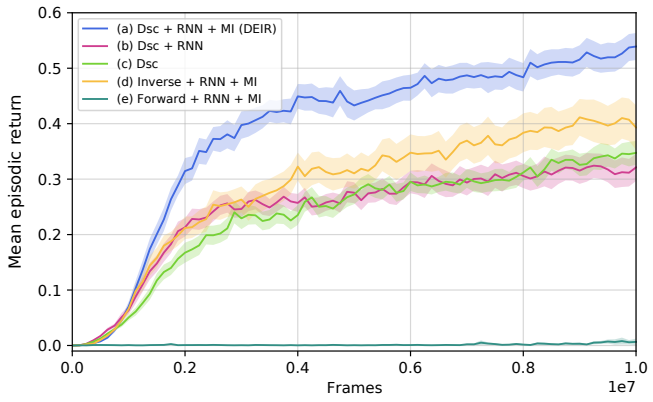


Figure 6: Results of ablation studies in an advanced *KeyCorridorS6R3* task. The effectiveness of the conditional mutual information (MI) can be confirmed by comparing (a), (b), (c). The difference between (a), (d), (e) shows the discriminator's importance.

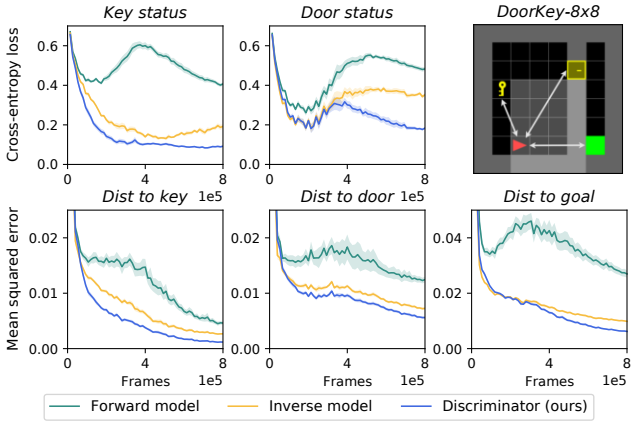


Figure 7: Validation losses of predicting temporal and spatial metrics using trajectory embeddings by three models. Upper-right figure exemplifies the objects (key, door, goal) related to the metrics. Upper: Whether the agent has picked up the key and opened the door. Lower: The agent's normalized distances to the key, door, and goal.

information term (see Figure 6(a) and (b)). As we can see, DEIR performed significantly better than the latter, which demonstrates the importance of our intrinsic reward design.

We utilize RNN in our model to capture temporal information in trajectories, which enables a more accurate representation to be learned. Thus, we further examine the effect of RNN on its own by training a separate agent with the discriminator only (see Figure 6(c)) and found that using RNN alone barely brings any benefit compared to Figure 6(b). Thus, we are confident that the mutual information scaling term indeed contributes to all of the performance improvements.

**Discriminative Model**

The performances of the DEIR agents driven by the inverse model and the forward model are shown in Figure 6(d) and (e), respectively. We applied the conditional mutual information term and RNN to both, and used the same tuned hyperparameters as in our previous experiments. Compared with Figure 6(a), our discriminative model-driven agent presented an evident advantage over the agent trained with the inverse model alone, while the forward model-driven agent completely failed to learn any meaningful policy in the task (due to the fact that the forward model is notorious for being weak to noise [Pathak *et al.*, 2017]), suggesting that to achieve an advanced performance, the discriminative model is indispensable for learning state representations in POMDPs. We believe the main contribution stems from the trajectory embeddings learned by the discriminative model, and we present further analysis results in Section 4.4 for more insights.

### 4.4 Effectiveness of Learned Embeddings

To obtain further insights into the impact of the learned embeddings on the final performance, we conducted an experiment to compare three model variants. Concretely, we trained a forward-only, an inverse-only, and a discriminative model using the same data sampled by a vanilla PPO agent in standard *DoorKey-8x8*, where an agent needs to find a key to open a door that leads to the goal in another room. Following Alain and Bengio's work [Alain and Bengio, 2016], we devise five auxiliary supervised learning tasks. Given the learned trajectory embeddings $e^{\text{TRAJ}}$, each task predicts an important temporal or spatial metric that is directly related to the game's progress or the agent's position. Technically, the ground-truth metrics are retrieved from the game's backend, and training stops after the agent reaches a near-optimal policy. The results in Figure 7 demonstrate that our discriminator-based method learns the most helpful embeddings for predicting important temporal and spatial metrics. This explains why it can benefit downstream objectives in RL, including exploration.

In comparison, embeddings from the forward and inverse models did not perform as well as ours, which is consistent with the findings in Figure 6. We hypothesize that the inverse model is less motivated to include historical data in embeddings because, by design, it can reliably infer actions merely from its always-factual inputs. On the other hand, the forward model relies too much on visual details, so its embeddings do not carry much information crucial to underlying tasks [Gulrajani *et al.*, 2017; Goyal *et al.*, 2017].

## 5 Conclusion

Training RL agents to explore effectively is a challenging problem, especially in environments with sparse rewards. A promising approach is to augment the extrinsic rewards with novelty-driven intrinsic rewards. However, focusing only on the novelty of observations is insufficient because an agent may incorrectly recognize the stochasticity in the environment's dynamics as novelties brought by its explorations. In this work, we proposed scaling the observation novelty with a conditional mutual information term that explicitly relates the agent's actions to the distances between observations, and learning a discriminative model that gives better intrinsic rewards. Compared with baselines, our method delivers outstanding performances in both standard and advanced versions of MiniGrid tasks. Also, it demonstrates general applicability to a variety of tasks with higher-dimensional inputs (such as those in ProcGen). As future work, we envision research on continuous action spaces and multi-agent settings.

# A Technical Appendix

## A.1 Intrinsic Reward Derivation

Given a pair of states $(s_t, s_i), \forall i \in [0, t)$, a pair of observations $(o_{t+1}, o_i)$, and an action $a_t$ from an episode's history, we optimize the agent's policy with an intrinsic reward

$$J = \mathrm{dist}(o_{t+1}, o_i) \cdot I\left(\mathrm{dist}(o_{t+1}, o_i); a_t | s_t, s_i\right). \quad \text{(A1)}$$

The first term of $J$ is the distance between observations $o_{t+1}$ and $o_i$, as used in previous studies [Blundell *et al.*, 2016; Pritzel *et al.*, 2017; Badia *et al.*, 2020]. The second term of $J$ is the conditional mutual information between $\mathrm{dist}(o_{t+1}, o_i)$ and $a_t$ given $s_t, s_i$. Our intuition behind such design is that the policy achieves novelty by maximizing the distance between observations $(o_{t+1}, o_i)$ *and* letting the distances be closely related to $a_t$, the action taken in $s_t$. This would efficiently eliminate the novelties rooted in the environment's stochasticity other than those brought by agent explorations.

The second term of $J$ can be derived as (denoting $D_{t+1,i} \triangleq \mathrm{dist}(o_{t+1}, o_i)$ for simplicity).

$$I(D_{t+1,i}; a_t | s_t, s_i) = \mathop{\mathbb{E}}_{s_t, s_i, a_t} [\mathrm{D_{KL}}(p(D_{t+1,i} | s_t, s_i, a_t) \| p(D_{t+1,i} | s_t, s_i))],$$

which is because

$$\begin{aligned} & I(X; Y|Z) \\ &= \mathop{\mathbb{E}}_{p(x,y,z)} \left[\log \frac{P(X,Y|Z)}{P(X|Z)P(Y|Z)}\right] \\ &= \mathop{\mathbb{E}}_{p(x,y,z)} \left[\log \frac{P(X|Y,Z)\cancel{P(Y|Z)}}{P(X|Z)\cancel{P(Y|Z)}}\right] \\ &= \mathop{\mathbb{E}}_{p(y,z)} \left[\mathbb{E}_{p(x|y,z)} \left[\log \frac{P(X|Y,Z)}{P(X|Z)}\right]\right] \\ &= \mathop{\mathbb{E}}_{p(y,z)} \left[\mathrm{D_{KL}}(P(X|Y=y, Z=z) \| P(X|Z=z))\right]. \end{aligned}$$

We seek to simplify $J$, which contains an intractable evidence term in the Kullback–Leibler (KL) divergence $D_{KL}$. To do so, we devise a surrogate function for the KL divergence by connecting it with total variation (TV). Denoting $P(x) = p(x|s_t, s_i, a_t)$ and $Q(x) = p(x|s_t, s_i)$, the Bretagnolle–Huber inequality [Bretagnolle and Huber, 1978] gives

$$\begin{aligned} & \mathrm{d_{TV}}(P, Q) \le \sqrt{1 - \exp(-\mathrm{D_{KL}}(P\|Q))} \\ \Leftrightarrow\ & \mathrm{D_{KL}}(P\|Q) \ge -\log(1 - \mathrm{d}^2_{\mathrm{TV}}(P, Q)), \end{aligned}$$

where $\mathrm{d_{TV}}(P, Q) = \frac{1}{2}\|P - Q\|_1$ is the total variation between $P$ and $Q$. Since $\mathrm{d_{TV}}(P, Q)$ monotonically increases w.r.t. to KL divergence $D_{KL}(P\|Q)$, we use $\mathrm{d_{TV}}$ as the surrogate function. Furthermore, given $s_t, s_i, a_t$ in any deterministic environment (including POMDPs), $s_{t+1}$, $o_{t+1}$, and $D_{t+1,i}$ are uniquely determined. $P$ is thus a unit impulse function that has the only non-zero value at $D_{t+1,i}$:

$$P(x) = \begin{cases} +\infty & x = D_{t+1,i} \\ 0 & x \ne D_{t+1,i} \end{cases}, \quad \int_{\mathcal{D}} P(x)dx = 1,$$

where $\mathcal{D}$ denotes the set of all possible observation distances. Also, by the definition of $Q$, we have $\int_{\mathcal{D}} Q(x)dx = 1$ and $Q(D_{t+1,i}) \le 1$. Given those properties and Scheffé's theorem [Tsybakov, 2008], we obtain

$$\mathrm{d_{TV}} = 1 - \int_{\mathcal{D}} \min\big(P(x), Q(x)\big)dx = 1 - Q(D_{t+1,i}).$$

We derive the KL divergence's lower bound using $\mathrm{d_{TV}}$, as

$$\begin{aligned} \mathrm{D_{KL}}(P\|Q) &\ge -\log(1 - \mathrm{d}^2_{\mathrm{TV}}(P, Q)) \\ &= -\log(1 - (1 - Q(D_{t+1,i}))^2) \\ &= -\log(2 \times Q(D_{t+1,i}) - Q(D_{t+1,i})^2) \\ &\ge -\log(2 \times Q(D_{t+1,i})) \\ &= -\log(2) - \log(Q(D_{t+1,i})), \end{aligned}$$

where $-\log(2)$ is constant, and we effectively maximize the last term $-\log\left(Q(D_{t+1,i})\right)$. Since $D_{t+1,i} \triangleq \mathrm{dist}(o_{t+1}, o_i)$ is the non-negative distance between two observations $o_{t+1}$ and $o_i$, and $\mathrm{dist}(o_{t+1}, o_i)$ is related to $\mathrm{dist}(s_t, s_i)$, we assume $D_{t+1,i}$ approximately follows an exponential distribution with a mean of the distance between its closest underlying states $s_t$ and $s_i$. The intuition here is that limited observability leads to similar observations with minor variations, especially during early training phases when exploration is crucial. The marginal distributions of observation distances observed in our experiments also confirmed this. By setting $D_{t+1,i}|s_t, s_i \sim \mathrm{Exp}(\lambda = 1/\mathrm{dist}(s_t, s_i))$ with an expected value of $1/\lambda = \mathrm{dist}(s_t, s_i)$, we obtain

$$\begin{aligned} & -\log\big(Q(D_{t+1,i})\big) \\ &\quad = -\log\left(\lambda \exp\left(-\lambda D_{t+1,i}\right)\right) \\ &\quad = -\log\Big(\frac{1}{\mathrm{dist}(s_t, s_i)} \exp\big(-\frac{\mathrm{dist}(o_{t+1}, o_i)}{\mathrm{dist}(s_t, s_i)}\big)\Big) \\ &\quad = \log(\mathrm{dist}(s_t, s_i)) + \frac{\mathrm{dist}(o_{t+1}, o_i)}{\mathrm{dist}(s_t, s_i)}. \end{aligned}$$

With all of the above, we finally define $J$'s lower bound as

$$J \ge \mathrm{dist}(o_{t+1}, o_i) \cdot \mathop{\mathbb{E}}_{s_t, s_i, a_i} \left(\log(\mathrm{dist}(s_t, s_i)) + \frac{\mathrm{dist}(o_{t+1}, o_i)}{\mathrm{dist}(s_t, s_i)}\right). \quad \text{(A2)}$$

Note that the full distributions of observations and states are difficult to sample. By keeping only the dominating term $\mathrm{dist}(o_{t+1}, o_i)/\mathrm{dist}(s_t, s_i)$ and relaxing the expectation to the minimum, we further simplify Equation A2 to

$$J \ge \min_i \frac{\mathrm{dist}^2(o_{t+1}, o_i)}{\mathrm{dist}(s_t, s_i)}, \quad \text{(A3)}$$

where $o_{t+1}, o_i$, and $s_t, s_i$ are random variables in Equations A1 and A2 but realized values in Equation A3. We find that Equation A3 is much simpler than Equation A2, yet delivers a performance that is just as good or even better. Equation A3 is equivalent to the intrinsic reward proposed in the main text (Equation 4).

## A.2 Implementation Details

Our implementations are based on Stable Baselines 3 [Raffin *et al.*, 2021] and the official code of existing methods (if available). Our source code is available at https://github.com/swan-utokyo/deir. More details about our algorithms, benchmarks, hyperparameters, network structures, and experimental results can be found at https://arxiv.org/abs/2304.10770.

## References


[Agarwal *et al.*, 2021] Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

[Alain and Bengio, 2016] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[Badia *et al.*, 2020] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martin Arjovsky, Alexander Pritzel, Andrew Bolt, et al. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, 2020.

[Baker *et al.*, 2019] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2019.

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

[Blundell *et al.*, 2016] Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z. Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control. *arXiv preprint arXiv:1606.04460*, 2016.

[Bretagnolle and Huber, 1978] Jean Bretagnolle and Catherine Huber. Estimation des densités: risque minimax. *Séminaire de probabilités de Strasbourg*, 12:342–363, 1978.

[Burda *et al.*, 2019] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *Seventh International Conference on Learning Representations*, 2019.

[Chevalier-Boisvert *et al.*, 2018] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018. Accessed: 2023-05-18.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

[Cobbe *et al.*, 2019] Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Procgen benchmark. https://github.com/openai/procgen, 2019. Accessed: 2023-05-18.

[Cobbe *et al.*, 2020] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International Conference on Machine Learning*, pages 2048–2056. PMLR, 2020.

[Cobbe *et al.*, 2021] Karl W. Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *International Conference on Machine Learning*, pages 2020–2027. PMLR, 2021.

[Espeholt *et al.*, 2018] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

[Goyal *et al.*, 2017] Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. In *Advances in neural information processing systems*, volume 30, 2017.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. PixelVAE: A latent variable model for natural images. In *International Conference on Learning Representations*, 2017.

[Hafner *et al.*, 2020] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Laskin *et al.*, 2020] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Mnih *et al.*, 2015] 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.

[Murphy, 2022] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

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

[Pathak *et al.*, 2017] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International*

*Conference on Machine Learning*, pages 2778–2787. PMLR, 2017.

[Pritzel *et al.*, 2017] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International Conference on Machine Learning*, pages 2827–2836. PMLR, 2017.

[Raffin *et al.*, 2021] 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.

[Raileanu and Fergus, 2021] Roberta Raileanu and Rob Fergus. Decoupling value and policy for generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 8787–8798. PMLR, 2021.

[Rosenblatt, 1958] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[Rumelhart *et al.*, 1986] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Internal Representations by Error Propagation*. MIT Press, 1986.

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

[Scott, 2010] Steven L. Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[Sutton and Barto, 2018] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 2nd edition, 2018.

[Tang *et al.*, 2017] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4–9, 2017.

[Tsybakov, 2008] Alexandre B. Tsybakov. Introduction to nonparametric estimation. In *Springer Series in Statistics*, 2008.

[Zhang *et al.*, 2021] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E. Gonzalez, and Yuandong Tian. NovelD: A simple yet effective exploration criterion. In *Advances in Neural Information Processing Systems*, 2021.