Decoupling Exploration and Exploitation in Reinforcement Learning

Lukas Schäfer [1] **Filippos Christianos** [1] **Josiah Hanna** [1 2] **Stefano V. Albrecht** [1]

Abstract

Intrinsic rewards are commonly applied to improve exploration in reinforcement learning. However, these approaches suffer from instability caused by non-stationary reward shaping and strong dependency on hyperparameters. In this work, we propose Decoupled RL (DeRL) which trains separate policies for exploration and exploitation. DeRL can be applied with on-policy and off-policy RL algorithms. We evaluate DeRL algorithms in two sparse-reward environments with multiple types of intrinsic rewards. We show that DeRL is more robust to scaling and speed of decay of intrinsic rewards and converges to the same evaluation returns than intrinsically motivated baselines in fewer interactions.

1. Introduction

Exploration is one of the essential challenges in reinforcement learning (RL). RL algorithms often use simple randomised methods, e.g. applying ϵ-greedy policies (Watkins, 1989) or adding random noise to continuous actions (Lillicrap et al., 2015). Such exploration techniques are inefficient on tasks where rewards are sparse. One category of exploration techniques which has been found to be particularly effective in sparse-reward environments are intrinsic rewards (Raileanu & Rocktäschel, 2020; Burda et al., 2018b; Pathak et al., 2017; Oudeyer et al., 2007; Chentanez et al., 2005). These can serve as strong incentives for exploration and assist training in hard exploration problems where a steady stream of rewards is missing.

Two common approaches to define intrinsic rewards r^i are self-supervised predictions in the environment (Raileanu & Rocktäschel, 2020; Burda et al., 2018b; Pathak et al., 2017; Schmidhuber, 1991) and computing (pseudo-) counts of states (Ostrovski et al., 2017; Tang et al., 2017; Strehl & Littman, 2008). These intrinsic rewards are added to the extrinsic reward r^e provided by the environment to define a combined reward signal $r = r^e + \lambda r^i$ with some weighting factor λ. However, optimising for this combined feedback introduces three challenges. (1) Intrinsic rewards are designed to incentivise exploration and naturally decay as more exploration has been completed. Therefore, intrinsic rewards are constantly changing. Such **non-stationary reward shaping** violates the Markov assumption and can cause major instabilities. (2) Careful selection of λ for **intrinsic reward scaling is essential** for their effective application. If intrinsic rewards are too large, they might heavily distort training and effectively introduce non-stationary noise to the optimisation. We show that small intrinsic rewards have no sufficient impact and do not incentivise exploration as intended. (3) Intrinsic rewards are designed to decay throughout training, but training is **sensitive to their speed of decay**. Similar to the scale of intrinsic rewards, we show that slowly decaying intrinsic rewards disrupt training whereas quickly vanishing intrinsic rewards have no sufficient impact on exploration.

These challenges lead to a significant dependency of intrinsic rewards on hyperparameters to determine their scale and decay. These dependencies make the application of intrinsic rewards highly task-dependent because the optimal scale and decay strongly depends on the scale of extrinsic rewards and required exploration in the respective task. All these properties make the practical application of such methods difficult (Taïga et al., 2019).

Motivated by these challenges and success in off-policy RL (Fujimoto et al., 2018; Haarnoja et al., 2018; Silver et al., 2014; Degris et al., 2012), we propose to separate the RL training into two separate policies. We train a *exploration policy* π_β for exploration with the combined signal of extrinsic and intrinsic rewards. Alongside, we train an *exploitation policy* π_e using only extrinsic rewards on the data collected by the exploration policy. We refer to this approach as **Decoupled RL** (DeRL)*. Using such decoupling addresses all the aforementioned challenges of previous application of intrinsic rewards. The exploration policy is

[1]School of Informatics, University of Edinburgh, Edinburgh, United Kingdom [2]To be joining the Computer Sciences Department, University of Wisconsin—Madison. Correspondence to: Lukas Schäfer <l.schaefer@ed.ac.uk>.

Unsupervised Reinforcement Learning (URL) Workshop in the 38^{th} International Conference on Machine Learning, 2021.

*We provide an open-source implementation of DeRL here: https://github.com/uoe-agents/derl

optimised using the combined objective of extrinsic and intrinsic rewards as in typical intrinsically motivated RL, but it is not required to be an effective policy for application in the task. Instead, π_β is only trained to generate data which can be used to train the exploitation policy. Our experiments show that DeRL is less sensitive to scale and speed of decay of intrinsic rewards and π_e is not affected by non-stationary reward shaping as it is only trained using extrinsic rewards.

We implement and evaluate on-policy and off-policy versions of DeRL with five types of intrinsic rewards in two learning environments that focus on exploration. The exploitation policy of several DeRL algorithms is able to converge to highest evaluation returns using fewer interactions in a series of tasks compared to traditional intrinsically motivated RL baselines. We also analyse the sensitivity of DeRL and RL baselines to the scale and the speed of decay of intrinsic rewards to verify the general dependency of these methods on the hyperparameters of intrinsic rewards and show that DeRL is more robust to varying hyperparameters. Such improved robustness and sample efficiency can justify the additional cost of training a second policy. However, we also observe that DeRL is only marginally more robust and still suffers from significant instabilities in the off-policy optimisation of the exploitation policy π_e in several tasks. We hypothesise that distribution shift caused by divergence of π_e and π_β leads to these instabilities, and propose future work to address these shortcomings.

2. Related Work

A variety of methods have been proposed to replicate the exploration incentive of curiosity for RL (Barto, 2013; Oudeyer & Kaplan, 2009; Oudeyer et al., 2008). The underlying idea behind these approaches is fairly simple: agents should be incentivised to explore novel or poorly understood parts of the environment. Therefore, an agent learns an intrinsic reward which rewards the agent for visiting novel states. Over time, the agent should become less "curious" with completed exploration and exploitation will gradually take over. There are two common branches of intrinsic rewards for exploration: (1) count-based and (2) prediction-based intrinsic rewards. Below, we will provide an overview over several of these approaches.

2.1. Count-based Intrinsic Rewards

Count-based intrinsic rewards are defined to be inverse proportional to the count of visitations of encountered states:

$$r_t^i := \frac{1}{\sqrt{N(s_t)}} \tag{1}$$

Thereby, agents are incentivised to visit states within the environment which are less frequently encountered. Likewise, agents are discouraged from visiting frequently encountered states which are deemed less valuable for exploration. While this approach is easily applicable in small, discrete state spaces, pseudo-counts have to be computed for large or continuous state spaces where encountering any state multiple times is rare. These pseudo-counts can be computed using various techniques to group similar states together.

Density models: Bellemare et al. (2016) and Ostrovski et al. (2017) achieve such grouping with a density model predicting the visitations of states. Such pseudo-counts considerably improved exploration in multiple Atari games.

Hash functions: Tang et al. (2017) propose to generalise pseudo-counts across similar states with a locality-sensitive hash function (Andoni & Indyk, 2008) for states which is more efficient to compute compared to deep density models. The paper specifically proposes the SimHash (Charikar, 2002) function measuring angular distance based on the sign of a randomised mapping.

2.2. Prediction-based Intrinsic Rewards

A separate approach defines intrinsic rewards using predictions in the environment. Schmidhuber (1991) proposed an intrinsic reward defined as the prediction error of predicting the next state given the current state and action. However, stochastic and thereby unpredictable dynamics within the environment would lead to the exploration signal constantly remaining high at the face of unpredictability. While a concept of boredom has been proposed to decay rewards for "not learnable" dynamics, this remains a major challenge of these approaches and is commonly called the "noisy TV problem" (Burda et al., 2018a).

Intrinsic curiosity module (ICM): Pathak et al. (2017) propose the intrinsic curiosity module to learn efficient state representations $\phi(s)$ and assign an intrinsic reward for the prediction error of the next state

$$r_t^i := \left(\widehat{\phi}(s_{t+1}) - \phi(s_{t+1})\right)^2 \tag{2}$$

where ϕ is a learned self-supervised state-representation function trained using an inverse-dynamics objective: given a representation of current state, $\phi(s_t)$, and next state, $\phi(s_{t+1})$, predict the applied action a_t. Through this feature representation, the model only learns to encode information of states which might be affected through the agent's actions. Using such representation, the prediction error is based on the prediction $\widehat{\phi}(s_{t+1})$ of the next state given current state s_t and applied action a_t.

Rewarding impact-driven exploration (RIDE): Raileanu & Rocktäschel (2020) propose rewarding impact-driven exploration which aims to reward the agent for applying actions which lead to significant change in the environment.

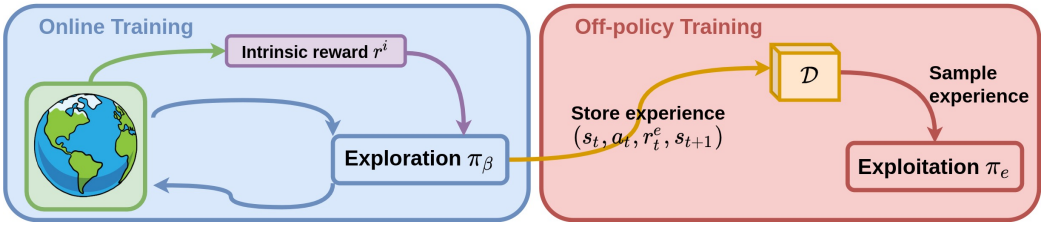


Figure 1: Visualisation of Decoupled Reinforcement Learning training loop.

Such impact is defined as the difference between embeddings of consecutive states, where the embedding function is trained using an inverse dynamics model identical to ICM (Pathak et al., 2017). In order to avoid the agent going back and forth between a group of states, an episodic state-count is added to the objective

$$r_t^i := \frac{\left(\phi(s_{t+1}) - \phi(s_t)\right)^2}{\sqrt{N_{ep}(s_{t+1})}} \tag{3}$$

where ϕ is the trained state embedding and N_{ep} represents the episodic (pseudo-) count of states.

Random network distillation (RND): Burda et al. (2018b) propose a prediction-based intrinsic reward for exploration based on two observations: (1) training and computation of complex models of dynamics is expensive and (2) static random feature representation is surprisingly effective for state predictions (Burda et al., 2018a). Random network distillation computes a representation of states trained to mimic a randomly initialised, fixed target representation

$$r_t^i := \left(\widehat{\phi}(s_t) - \phi(s_t)\right)^2 \tag{4}$$

where ϕ is a randomised state representation function and $\widehat{\phi}$ is trained to represent ϕ.

2.3. Off-Policy Reinforcement Learning

RL trains a policy to collect data and to maximise the extrinsic returns in the environment. Exploration and exploitation typically apply different policies, but these are often derived from each other, e.g. a greedy policy for exploitation. We propose to separate the training of two distinct policies for exploration and exploitation, but training the exploitation policy using experience generated by the exploration policy requires off-policy RL. Off-policy RL is concerned with the optimisation of a policy π_e using experience generated within the environment by following a separate behaviour policy π_β. Separating exploration and exploitation enables the exploration policy to be trained for exploration without modifying the training objective of the exploitation policy. However, one major challenge of off-policy RL is *distribution shift*, caused by differences of both policies (Fujimoto et al., 2018). Such differences cause the state distributions and action distributions of both policies to diverge and can cause significant instabilities during training.

2.4. Unbiased Task Policy

Whitney et al. (2021) recently proposed to separately train an exploration and task policy using off-policy RL. In their work, they train a task policy and concurrently train an exploration policy using only intrinsic rewards. Their behaviour policy is defined as a factored policy of both the task and exploration policy. In their work, they solely focused on off-policy task policies trained using soft Double-DQN (van Hasselt et al., 2016) and further optimisations focused on fast adaptation of the exploration policy. Intrinsic rewards are defined using a simple count-based definition which is also used for optimisitic initialisation (Rashid et al., 2020) but no further intrinsic reward definitions are considered. In contrast, we consider a range of different intrinsic rewards and find that on-policy training of the task policy leads to better results with off-policy correction.

3. Decoupled Reinforcement Learning

In this work, we propose to decouple exploration and exploitation into two separate policies to improve sample efficiency and overcome sensitivity to hyperparameters of intrinsic rewards. We train an exploration policy π_β with the intent to explore the environment. Using the data collected by the exploration policy, a separate exploitation policy π_e is trained, as visualised in Figure 1.

Formally, an agent trains a exploration policy π_β which is trained to maximise the sum of intrinsic and extrinsic rewards to achieve exploration,

$$\pi_\beta \in \arg\max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t^e + \lambda r_t^i\right) \mid a_t \sim \pi(s_t)\right] \tag{5}$$

$$= \arg\max_\pi \mathbb{E}\left[G_t^{e+i} \mid a_t \sim \pi(s_t)\right] \tag{6}$$

with G^{e+i} denoting the discounted returns computed using the combination of extrinsic and intrinsic rewards with scaling factor λ and discount factor $\gamma \in [0, 1)$. During training of π_β, experience samples $(s_t, a_t, r_t^e, s_{t+1})$ with extrinsic rewards are collected in a set $\mathcal{D}$.

Algorithm 1 Decoupled On-Policy Learning

Initialise: parameters ϕ, θ and π_β
$\mathcal{D} \leftarrow \emptyset$
$i \leftarrow 0$
for ep $= 0, \dots N_{ep}$ **do**
  $a_t \sim \pi_\beta(s_t)$
  $s_{t+1}, r^e_t \leftarrow$ environment step with a_t
  Update π_β
  $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, r^e_t, s_{t+1})$
  $i \leftarrow i + 1$
  if $i \mod T_{Dec} = 0$ **then**
    Update ϕ with Equation (9) and $\mathcal{D}$
    Update θ with Equation (12) and $\mathcal{D}$
    $\mathcal{D} \leftarrow \emptyset$
  end if
end for

In addition to this typical intrinsically motivated RL, we train a separate exploitation policy π_e to maximise only expected cumulative extrinsic rewards using experience accumulated in $\mathcal{D}$ with G^e_t denoting discounted extrinsic returns.

$$\pi_e \in \arg\max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^e_t \mid (s_t, a_t, r^e_t, s_{t+1}) \sim \mathcal{D}\right] \quad (7)$$

$$= \arg\max_\pi \mathbb{E}\left[G^e_t \mid (s_t, a_t, r^e_t, s_{t+1}) \sim \mathcal{D}\right] \quad (8)$$

Both exploration π_β and exploitation policy π_e can be trained using any RL algorithm given the defined objectives. Generally, the exploration policy π_β is optimised as typical RL with intrinsic rewards, whereas π_e is trained every T_{Dec} timesteps on experience from $\mathcal{D}$ which is generated by π_β's interaction in the environment. Note that $\mathcal{D}$ only contains extrinsic rewards and is off-policy data for the optimisation of π_e as it was generated by following π_β. Below, we propose two methods to apply such decoupled RL in an on-policy and off-policy fashion.

3.1. Decoupled On-Policy Reinforcement Learning

In order to apply on-policy RL algorithms to train π_e using $\mathcal{D}$, off-policy correction can be applied to account for differences in trajectory distributions of both π_e and π_β.

One technique for off-policy correction is *importance sampling* (IS). In the following, we assume π_e is trained using an on-policy actor-critic RL algorithm with state value function V, parameterised by θ, and policy π_e, parameterised by ϕ. The latter are optimised by minimising the actor loss given in Equation (9)

$$\mathcal{L}(\phi) = \mathbb{E}\Big[-\rho(A_t|S_t) \log \pi_e(A_t|S_t;\phi)\, \widehat{G}(S_t) \mid (S_t, A_t, R^e_t, S_{t+1}) \sim \mathcal{D}\Big] \quad (9)$$

with bootstrapped return estimates $\widehat{G}(S_t)$

$$\widehat{G}(S_t) = (R^e_t + \gamma V(S_{t+1};\theta) - V(S_t;\theta)) \quad (10)$$

Algorithm 2 Decoupled Off-Policy Learning

Initialise: parameters θ and π_β
$\mathcal{D} \leftarrow \emptyset$
$i \leftarrow 0$
for ep $= 0, \dots N_{ep}$ **do**
  $a_t \sim \pi_\beta(s_t)$
  $s_{t+1}, r^e_t \leftarrow$ environment step with a_t
  Update π_β
  $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, r^e_t, s_{t+1})$
  $i \leftarrow i + 1$
  if $i \mod T_{Dec} = 0$ **then**
    Update θ with Equation (13) and $\mathcal{D}$
  end if
end for

and IS weights $\rho(A_t|S_t)$.

$$\rho(A_t|S_t) = \frac{\pi_e(A_t|S_t;\phi)}{\pi_\beta(A_t|S_t)} \quad (11)$$

Similarly, the value loss for π_e using IS weights can be defined as follows:

$$\mathcal{L}(\theta) = \mathbb{E}\Big[\rho(A_t|S_t)\ \left(V(S_t;\theta) - (R^e_t + \gamma V(S_{t+1};\theta))\right)^2 \mid (S_t, A_t, R^e_t, S_{t+1}) \sim \mathcal{D}\Big] \quad (12)$$

IS weights ρ can cause instabilities during off-policy training through exploding weights when $\pi_e(A_t|S_t;\phi) \gg \pi_\beta(A_t|S_t)$ or vanishing weights for $\pi_e(A_t|S_t;\phi) \ll \pi_\beta(A_t|S_t)$. In particular, such instabilities occur when one policy assigns approximately zero probability for some action. Various techniques have been proposed to address such instabilities, including clipping of importance weights (Munos et al., 2016; Espeholt et al., 2018) to minimise vanishing or exploding gradients. The pseudocode for the optimisation of π_e using on-policy RL can be found in Algorithm 1.

3.2. Decoupled Off-Policy Reinforcement Learning

Instead of optimising π_e using on-policy RL algorithms with off-policy correction such as importance sampling, any off-policy RL algorithm can be applied without the need for any correction. In this work, we consider optimising π_e using off-policy Deep Q-Networks (DQN) (Mnih et al., 2015). For DQN optimisation, the following Q-learning loss is minimised

$$\mathcal{L}(\theta) = \mathbb{E}\Big[\big(Q(S_t, A_t;\theta) - \bar{Q}(S_t)\big)^2 \mid (S_t, A_t, R^e_t, S_{t+1}) \sim \mathcal{D}\Big] \quad (13)$$

with target Q-values

$$\bar{Q}(S_t) = (R^e_t + \gamma \max_{a'} Q(S_{t+1}, a';\bar{\theta})) \quad (14)$$

and $\bar{\theta}$ denoting the parameters of the periodically updated target network. Pseudocode for off-policy optimisation of π_e can be found in Algorithm 2. Note that $\mathcal{D}$ is only used for a single update in decoupled on-policy learning, whereas in off-policy optimisation $\mathcal{D}$ represents a replay buffer (Lin, 1992) which is continually filled with experience.

4. Evaluation

We evaluate on- and off-policy DeRL in two learning environments with a variety of RL algorithms and intrinsic rewards. In particular, we investigate the following two hypotheses: (1) DeRL leads to higher returns or demonstrates improved sample-efficiency, converging to comparable returns than intrinsically motivated RL baselines in fewer environment interactions, and (2) DeRL is more robust than intrinsically motivated RL baselines to varying scale λ and speed of decay of intrinsic rewards.

4.1. Intrinsic Rewards

Count-based: We consider two count-based intrinsic rewards. **Dict-Count** directly stores and increments state occurrences in a table. **Hash-Count** additionally groups states using the SimHash function (Tang et al., 2017) before computing intrinsic rewards following Equation (1).

Prediction-based: Besides count-based intrinsic exploration definitions, we consider **ICM** (Pathak et al., 2017), **RND** (Burda et al., 2018b), and **RIDE** (Raileanu & Rocktäschel, 2020) as prediction-based approaches. ICM and RIDE both use the same inverse dynamics optimisation to compute state-representations and define intrinsic rewards based on next state prediction error and consecutive state differences, respectively. In contrast, RND initialises a random state representation and trains a network to mimic such representation. For more details, see Section 2.2.

4.2. Algorithms

Baselines: As baselines, we consider on-policy RL algorithms Advantage Actor-Critic (A2C) (Mnih et al., 2016) and Proximal Policy Optimisation (PPO) (Schulman et al., 2017). Both algorithms are trained using the combined reward $r_t = r_t^e + \lambda r_t^i$ with some weighting factor λ and intrinsic reward definition as given above.

DeRL: For our decoupled RL optimisation, we consistently train π_β using A2C as we found it to be more robust than PPO. We train π_β using Dict-Count and ICM as intrinsic rewards. For the optimisation of π_e, we consider A2C and PPO for on-policy DeRL and Deep Q-Networks (DQN) (Mnih et al., 2015) for off-policy DeRL. We refer to these algorithms as DeA2C, DePPO and DeDQN.

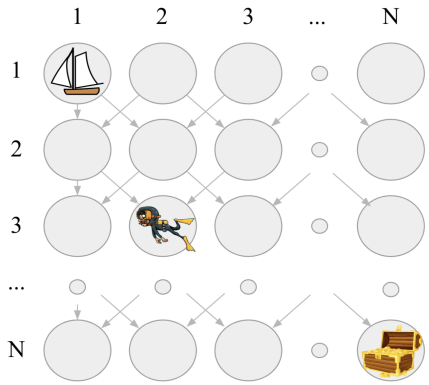


Figure 2: DeepSea environment, from Osband et al. (2020).

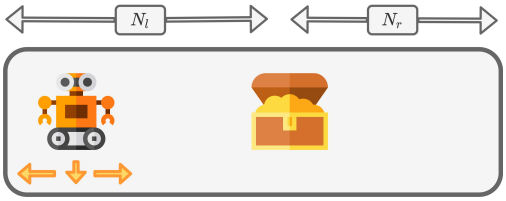


Figure 3: Hallway environment.

4.3. Environments

DeepSea is an environment proposed as part of the Behaviour Suite (Bsuite) for RL (Osband et al., 2020), visualised in Figure 2. The environment targets the challenge of exploration and represents a $N \times N$ grid where the agent starts in the top left and has to reach a goal in the bottom right location. At each timestep, the agent moves one row down and can choose one out of two actions. For each state, both actions are randomly assigned to left and right movement. The agent observes the current location as a 2D one-hot encoding and receives a small negative reward of $\frac{-0.01}{N}$ for moving right and 0 reward for moving left. Additionally, the agent receives a reward of $+1$ for reaching the goal and the episode ends after N timesteps. The difficulty of the exploration in DeepSea can be adjusted using N: the larger N, the harder it becomes for the agent to reach the goal location for optimal returns of 0.99. We evaluate all algorithms in the DeepSea task for $N \in \{10, 14, 20, 24, 30\}$.

Hallway is a new environment proposed as part of this work to target exploration, visualised in Figure 3. In this navigation task, similar to DeepSea, an agent is located in a hallway starting on the left. A goal can be reached by moving N_l cells to the right. In contrast to DeepSea, the goal is not necessarily located at the right end of the hallway, but there might be further N_r empty cells to the right of the goal location. At each timestep, the agent can choose between three actions: move left, stay or move right. The agent receives a reward of -0.01 for moving right or stay

Table 1: Average evaluation returns and a single standard deviation in all DeepSea and Hallway tasks over 100,000 episodes. The highest achieved returns in each task are highlighted in bold together with all returns within a single standard deviation. For DeRL evaluations are executed using the exploitation policy.

Alg	DeepSea 10	DeepSea 14	DeepSea 20	DeepSea 24	DeepSea 30	Hallway 10-0	Hallway 10-10	Hallway 20-0	Hallway 20-20	Hallway 30-0	Hallway 30-30
A2C	$\mathbf{0.93 \pm 0.22}$	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.67 ± 0.05	0.50 ± 0.05	0.52 ± 0.02	0.50 ± 0.01	0.28 ± 0.08	0.42 ± 0.08
A2C Dict-Count	$\mathbf{0.98 \pm 0.09}$	$\mathbf{0.94 \pm 0.16}$	$\mathbf{0.95 \pm 0.15}$	0.11 ± 0.15	-0.01 ± 0.00	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.02}$	0.55 ± 0.03	$\mathbf{0.55 \pm 0.06}$	-0.33 ± 0.14	-0.06 ± 0.07
A2C Hash-Count	$\mathbf{0.98 \pm 0.09}$	$\mathbf{0.96 \pm 0.15}$	0.39 ± 0.14	$\mathbf{0.53 \pm 0.12}$	-0.01 ± 0.00	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.02}$	0.56 ± 0.02	$\mathbf{0.55 \pm 0.05}$	-0.34 ± 0.15	-0.13 ± 0.11
A2C ICM	0.86 ± 0.21	0.69 ± 0.31	0.54 ± 0.23	$\mathbf{0.46 \pm 0.30}$	$\mathbf{0.08 \pm 0.12}$	0.62 ± 0.17	0.56 ± 0.18	0.26 ± 0.12	$\mathbf{0.78 \pm 0.27}$	$\mathbf{1.16 \pm 0.46}$	$\mathbf{0.64 \pm 0.38}$
A2C RND	-0.01 ± 0.00	0.19 ± 0.02	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.12 ± 0.02	-0.06 ± 0.00	-0.20 ± 0.01	-0.24 ± 0.00	-0.24 ± 0.00	-0.12 ± 0.00
A2C RIDE	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.85 \pm 0.04}$	$\mathbf{0.85 \pm 0.02}$	$\mathbf{0.70 \pm 0.00}$	$\mathbf{0.62 \pm 0.00}$	0.37 ± 0.04	0.28 ± 0.08
PPO	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO Dict-Count	0.86 ± 0.11	0.70 ± 0.17	0.57 ± 0.25	0.17 ± 0.18	$\mathbf{0.20 \pm 0.15}$	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
PPO Hash-Count	0.87 ± 0.08	0.77 ± 0.13	0.34 ± 0.14	0.28 ± 0.20	$\mathbf{0.12 \pm 0.13}$	0.39 ± 0.11	0.17 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO ICM	0.88 ± 0.15	0.28 ± 0.17	0.00 ± 0.03	0.12 ± 0.17	0.00 ± 0.03	0.05 ± 0.15	0.16 ± 0.17	0.02 ± 0.16	0.08 ± 0.19	-0.04 ± 0.08	-0.02 ± 0.14
PPO RND	0.23 ± 0.14	0.15 ± 0.08	-0.01 ± 0.00	0.00 ± 0.00	-0.01 ± 0.00	-0.04 ± 0.04	-0.05 ± 0.09	-0.21 ± 0.06	-0.17 ± 0.09	-0.27 ± 0.10	-0.27 ± 0.11
PPO RIDE	0.70 ± 0.09	0.00 ± 0.00	0.00 ± 0.02	-0.01 ± 0.00	-0.01 ± 0.00	-0.10 ± 0.03	0.14 ± 0.10	-0.21 ± 0.03	-0.08 ± 0.08	-0.32 ± 0.04	-0.29 ± 0.08
DeA2C Dict-Count	$\mathbf{0.98 \pm 0.10}$	0.65 ± 0.23	0.42 ± 0.16	0.07 ± 0.10	$\mathbf{0.09 \pm 0.08}$	$\mathbf{0.84 \pm 0.07}$	$\mathbf{0.84 \pm 0.08}$	0.42 ± 0.02	$\mathbf{0.70 \pm 0.01}$	0.55 ± 0.00	0.22 ± 0.02
DeA2C ICM	0.87 ± 0.19	0.29 ± 0.23	0.28 ± 0.24	0.08 ± 0.14	$\mathbf{0.05 \pm 0.11}$	0.77 ± 0.18	0.80 ± 0.17	0.44 ± 0.15	$\mathbf{0.53 \pm 0.20}$	0.52 ± 0.34	$\mathbf{0.97 \pm 0.51}$
DePPO Dict-Count	0.51 ± 0.23	$\mathbf{0.92 \pm 0.18}$	-0.01 ± 0.01	$\mathbf{0.63 \pm 0.27}$	-0.01 ± 0.00	0.73 ± 0.10	$\mathbf{0.84 \pm 0.06}$	0.56 ± 0.01	$\mathbf{0.55 \pm 0.04}$	-0.20 ± 0.17	-0.06 ± 0.07
DePPO ICM	0.65 ± 0.22	0.63 ± 0.48	0.00 ± 0.02	-0.01 ± 0.00	0.00 ± 0.00	0.82 ± 0.11	0.82 ± 0.11	0.64 ± 0.16	$\mathbf{0.57 \pm 0.07}$	-0.01 ± 0.25	0.26 ± 0.06
DeDQN Dict-Count	$\mathbf{0.98 \pm 0.10}$	$\mathbf{0.95 \pm 0.17}$	$\mathbf{0.87 \pm 0.20}$	$\mathbf{0.53 \pm 0.27}$	$\mathbf{0.10 \pm 0.10}$	-0.13 ± 0.04	-0.15 ± 0.04	-0.05 ± 0.05	-0.12 ± 0.08	-0.17 ± 0.07	-0.10 ± 0.06
DeDQN ICM	$\mathbf{0.94 \pm 0.20}$	0.59 ± 0.40	0.36 ± 0.24	0.24 ± 0.25	$\mathbf{0.05 \pm 0.12}$	-0.09 ± 0.09	0.04 ± 0.18	-0.11 ± 0.09	-0.19 ± 0.08	-0.26 ± 0.08	-0.19 ± 0.08

and 0 reward for moving left. Additionally, the episode ends after $2N_l$ steps and the agent receives a reward of $+1$ for reaching the goal for the first time and every time it stays at the goal location for 10 steps. Therefore, the optimal behaviour is to move to the goal location and stay there for the remaining timesteps of the episode to collect further reward. We hypothesise that Hallway tasks, in particular with $N_r > 0$, exploration through intrinsic rewards have to be carefully balanced because exploration incentives and extrinsic rewards in Hallway are not aligned. In contrast, receiving extrinsic reward in DeepSea requires the agent to reach states at the end of the environment, so intrinsic rewards for exploration strongly align with extrinsic rewards from the environment. We evaluate all algorithms in the Hallway environment with $N_l \in \{10, 20, 30\}$ and N_r either being 0 or equal to N_l.

4.4. Implementation Details

We compute n-step returns for reduced bias of value estimates in all algorithms. On-policy training uses four parallel, synchronous environments and an additional entropy regularisation term in the policy loss (Mnih et al., 2016). Double-DQN (van Hasselt et al., 2016) targets are computed for DQN. For details on the conducted gridsearch for hyperparameter tuning as well as all hyperparameters used throughout experiments, see Appendix A.

We train all algorithms for 100,000 episodes and evaluate every 1,000 episodes for a total of 100 evaluations throughout training by applying the greedy (evaluation) policy in the respective task for 8 episodes. Reported results are averaged across five random seeds with shading indicating a single standard deviation. Optimal returns are indicated using a dashed horizontal line. A weighting factor of $\lambda = 1$ is used for the combined reward signal unless stated otherwise.

5. Results

Table 1 shows average evaluation returns of all algorithms across all DeepSea and Hallway tasks. Returns are averaged across all 100 evaluations to indicate achieved returns as well as sample efficiency. Additionally, we include plots of normalised returns, tables showing the maximum achieved evaluation returns at any evaluation and learning curves for each individual task in Appendix B.

Promising results: In all DeepSea tasks, DeDQN is never outperformed with respect to average evaluation returns. In the majority of these tasks, it converges to similar returns and achieves comparable sample effiency to the best performing baselines. DeA2C and DePPO demonstrate similar returns and sample efficiency in some of these tasks (see Figures 9a and 9d). In harder Hallway tasks with $N_l = 20, N_r = 0$ and $N_l = N_r = 30$ (visualised in Figures 10b and 10f), the exploitation policies of DeA2C and DePPO converge to highest returns and are shown to be slightly more sample efficient reaching returns after fewer episodes of training compared to the best performing baselines. Generally, we can see that DeA2C learns the optimal policy in the majority of Hallway tasks for some of the five executed runs, but fails to converge to such behaviour consistently. Instead, the majority of baselines and some DeRL runs learn to reach the goal but move back and forth between the goal and its left neighboured cell. Presumably, consistently staying at the goal is rarely discovered due to the small negative punishment of staying at a cell.

In DeepSea and Hallway, we find DeA2C and DePPO to converge comparably with DePPO reaching higher returns quicker in several DeepSea and Hallway tasks (see Figures 9j and 10e). We believe training the exploitation policy using PPO can be beneficial when trained with off-policy data due to trust region optimisation applied by PPO (Schulman et al., 2015; 2017). This confines the policy change in a single optimisation step by constraining divergence of

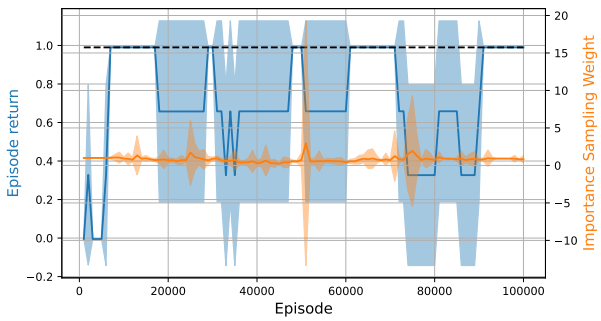


Figure 4: Evaluation returns and IS weights for DeA2C Dict-Count in DeepSea $N = 10$.

behaviour and trained policy. In our case, DePPO uses a clipping approximation of the KL-divergence between π_β and π_e to enforce both policies to not diverge too far from each other. Such an optimisation constraint seems to assist training of π_e by limiting the divergence of both policies.

Limitations and challenges: However, we also observe several remaining limitations and cases of failure for DeRL algorithms. DeDQN achieves low returns in the Hallway environment compared to both on-policy DeA2C and DePPO despite its off-policy training. Also, significant variance can be observed for baselines and DeRL algorithms in harder DeepSea and most Hallway tasks. Off-policy optimisation is theoretically independent of the policy generating training samples and in DeA2C and DePPO IS weight correction is applied to correct for the off-policy training data. However, we believe distribution shift (Fujimoto et al., 2018) is causing instabilities when optimising the exploitation policy from data generated by π_β. Figure 4 visualises unstable IS weights for DeA2C in the DeepSea task with $N = 10$ averaged over three seeds. These appear to correlate with some of the noticeable drops in returns throughout training indicating the negative impact of divergence of exploration and exploitation policies on RL training of π_e. Even when applying Retrace(λ) (Munos et al., 2016) to clip IS weights, similar results could be observed.

6. Hyperparameter Sensitivity

In order to investigate the impact of DeRL on the robustness to varying scale and speed of decay of intrinsic rewards, we train all baselines and decoupled algorithms on the combined objective $r = r^e + \lambda r^i$ in DeepSea $N = 10$ and Hallway $N_l = N_r = 10$ with varying λ and speeds of decay. All results with bar plots and tables showing maximum and average returns across varying hyperparameter values can be found in Appendix C.

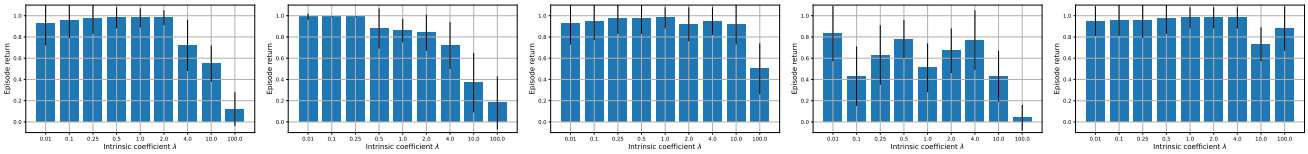
Scale of intrinsic rewards: We consider $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$ to analyse the sensitivity to varying scale of intrinsic rewards. Figure 5 shows average evaluation returns for all values of λ for baselines and DeRL with Dict-Count intrinsic rewards. Returns are averaged across three seeds before the mean and standard deviation across all 100 evaluation returns is computed. In DeepSea $N = 10$, similar sensitivity to varying values of λ can be observed for most algorithms. DeRL, in particular DeA2C and DeDQN, exhibit slightly improved robustness to varying values of λ compared to the baselines. In Hallway, all algorithms exhibit larger variance for varying λ with no significant learning being observed for large or small values of λ with DeA2C and DePPO demonstrating slightly more robustness. Overall, these results indicate the dependence of these methods on careful selection of λ. Even small deviations of λ can make the difference between learning and not learning at all.

Decay of intrinsic rewards: We also investigate the sensitivity of intrinsically motivated baselines and DeRL algorithms to the speed of decay of intrinsic rewards in both Hallway and DeepSea. For count-based intrinsic rewards, the speed of decay can be determined by the increment of the state count $N(s)$. For a sensitivity analysis, we consider increments $\{0.01, 0.1, 0.2, 1.0, 5.0, 10.0, 100.0\}$. For deep prediction-based intrinsic rewards the speed of decay is determined by their learning rates. We consider learning rates $\{1e^{-9}, 1e^{-8}, 2e^{-8}, 1e^{-7}, 5e^{-7}, 1e^{-6}, 1e^{-5}, 1e^{-4}, 1e^{-3}\}$. Figure 6 shows average evaluation returns for all considered count increments for baselines and DeRL with Dict-Count intrinsic rewards for both DeepSea $N = 10$ and Hallway $N_l = N_r = 10$. A2C appears slightly more robust to varying speed of decay in DeepSea $N = 10$ compared to PPO. DePPO demonstrates very similar returns for all decay values, but DeA2C and DeDQN are again shown to be the most robust algorithms exhibiting high evaluation returns for all considered values. Similarly as for λ sensitivity, we observe very significant dependency on the speed of decay in the Hallway task with DeA2C and DePPO exhibiting improved robustness to varying values. However, all DeRL algorithms and baselines appear highly dependent on the speed of decay of intrinsic rewards.

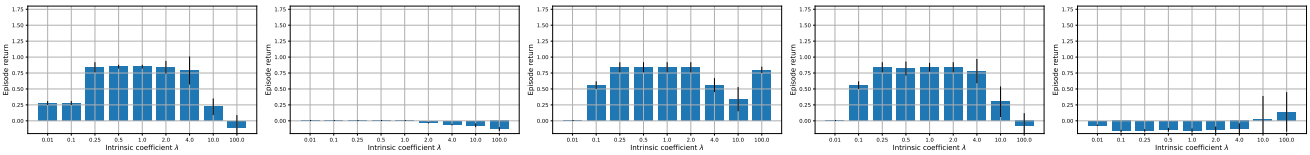
7. Future Work

This work serves as an initial investigation into the feasibility of decoupling exploration and exploitation in RL. However, there are two significant additions which we identified but did not address yet and require further investigation.

Minimising Distribution Shift: Training π_e using on-policy RL with IS correction still seems to suffer from significant variability. Based on our experiments with varying IS weights (see Figure 4), we believe that distribution shift of π_β and π_e is a cause for such variability. Therefore, we

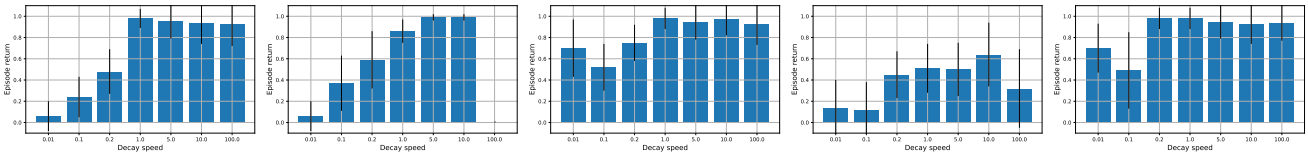


(a) A2C Dict-Count (b) PPO Dict-Count (c) DeA2C Dict-Count (d) DePPO Dict-Count (e) DeDQN Dict-Count

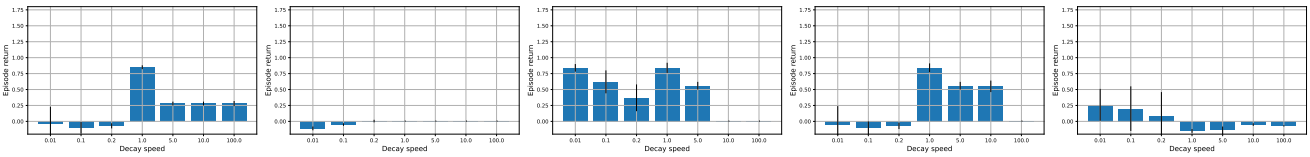


(f) A2C Dict-Count (g) PPO Dict-Count (h) DeA2C Dict-Count (i) DePPO Dict-Count (j) DeDQN Dict-Count

Figure 5: Average evaluation returns for A2C, PPO and DeRL with Dict-Count in DeepSea 10 (upper row) and Hallway $N_l = N_r = 10$ (lower row) with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.



(a) A2C Dict-Count (b) PPO Dict-Count (c) DeA2C Dict-Count (d) DePPO Dict-Count (e) DeDQN Dict-Count



(f) A2C Dict-Count (g) PPO Dict-Count (h) DeA2C Dict-Count (i) DePPO Dict-Count (j) DeDQN Dict-Count

Figure 6: Average evaluation returns for baselines and DeRL with Dict-Count in DeepSea 10 (upper row) and Hallway $N_l = N_r = 10$ (lower row) with varying count increments in $\{0.01, 0.1, 0.2, 1.0, 5.0, 10.0, 100.0\}$.

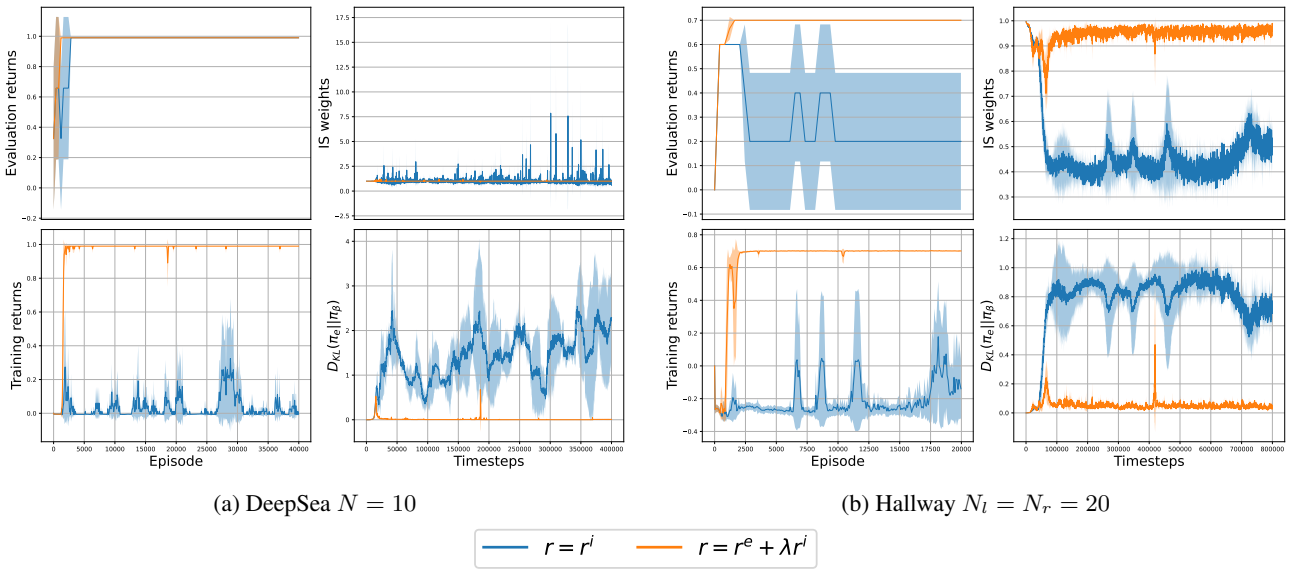
believe that the application of divergence constraints, proposed in the literature of offline RL (Levine et al., 2020), might be a valuable technique to minimise distribution shift. These auxiliary objectives are introduced to the optimisation and would enforce π_e and π_β to not diverge significantly by introducing a term $\alpha D\left(\pi_e(\cdot|S_t), \pi_\beta(\cdot|S_t)\right)$ to the optimisation loss. This term is based on a distance measure D between the distribution of policies π_β and π_e and some weighting hyperparameter α. Commonly-used distance measures are the maximum mean discrepancy (MMD) (Gretton et al., 2012), computed based on drawn samples from both policies, and the Kullback-Leibler (KL) divergence, which has been applied in offline RL (Jaques et al., 2019). The resulting regulariser can be written as

$$D_{KL}(\pi_e(\cdot|S_t), \pi_\beta(\cdot|S_t)) = \mathbb{E}_{A\sim\pi_e(\cdot|S)}\left[\log \pi_e(A|S_t) - \log \pi_\beta(A|S_t)\right] \quad (15)$$

For more distance measure candidates between two policies, see Wu et al. (2019) which also found most these metrics to perform comparably. Such divergence constraints have significant parallels to trust region optimisation (Schulman et al., 2015; 2017) as applied in DePPO for the optimisation of π_e and outlined in Section 5 which computes an approximation of the KL-divergence of π_β and π_e.

In DeRL, divergence constraints can be directly applied to the optimisation of either the exploration π_β or exploitation policy π_e. We can choose to keep π_β close to π_e and likewise can enforce π_e to stay close to π_β. We hypothesise, that adding such a constraining regularisation to the optimisation of π_β to enforce the exploration policy to stay close to the exploitation policy might be ideal as it does not interfere with the exploitation policy itself while still leveraging improved exploration of π_β. However, both directions as well as a combination of both constraints should be considered.

Only intrinsically rewarded exploration policy: The exploration policy π_β is solely trained to generate training data $\mathcal{D}$ for the optimisation of π_e. We hypothesise that it might be



(a) DeepSea $N = 10$

(b) Hallway $N_l = N_r = 20$

Figure 7: Preliminary results for training of DeA2C with Dict-Count in (a) DeepSea 10 and (b) Hallway $N_l = N_r = 20$ with the exploration policy being trained on the sum of intrinsic and extrinsic rewards or only intrinsic rewards. Training and evaluation returns are achieved using the exploration and exploitation policies, respectively.

beneficial to optimise π_β using only intrinsic rewards. Such optimisation would likely lead to more robustness to hyperparameters of intrinsic rewards as they would not be combined with extrinsic rewards of the environment. However, optimisation of π_β without extrinsic rewards might cause further divergence of π_β and π_e. We conducted preliminary experiments, training DeA2C with Dict-Count intrinsic rewards in both DeepSea 10 and Hallway $N_l = N_r = 20$ for 40,000 and 20,000 episodes, respectively, with the exploration policy being optimised using either the sum of extrinsic and intrinsic rewards or only using intrinsic rewards. Figure 7 shows that training the exploration policy with only intrinsic rewards does lead to distribution shift with both policies diverging more significantly as seen in IS weights and the KL divergence $D_{KL}(\pi_e||\pi_\beta)$. Training of the exploitation policy appears to suffer from such differences in the more challenging Hallway task, but in DeepSea the exploitation policy was successfully trained to solve the task despite the exploration policy never reaching high returns. These results indicate the feasibility of training π_β using only intrinsic rewards but also highlight remaining challenges due to diverging policies. We hypothesise that the addition of divergence constraints to the optimisation of the exploitation and exploration policies might address these challenges and make optimisation of π_β using only intrinsic rewards effective beyond simple tasks.

8. Conclusion

In this work, we propose Decoupled RL (DeRL) which separates exploration and exploitation into two separate policies. The exploration policy is optimised with additional intrinsic rewards to incentivise exploration and the exploitation policy is trained using only extrinsic rewards from data generated by the exploration policy. We believe such decoupling is primarily beneficial when tuning the exploration is challenging or heavy exploration is required which would typically dominate any extrinsic rewards received by the environment. On-policy and off-policy versions of DeRL are formulated and evaluated in two sparse-reward environments. We observe that DeRL demonstrates improved sample efficiency in some tasks by reaching high returns with fewer interactions in the environment. Our results also demonstrate that intrinsically motivated RL is highly dependent on careful hyperparameter tuning of intrinsic rewards with DeRL exhibiting slightly improved robustness to varying scale and speed of decay of intrinsic rewards. However, DeRL still seems to suffer from drops in performance and instability which correlate with divergence of the exploration and exploitation policies. Further research is proposed to address such distribution shift of both policies using divergence constraints similar to trust region optimisation. Lastly, the feasibility of training the exploration policy only using intrinsic rewards is demonstrated which serves as a second opportunity for future work.

References

Andoni, A. and Indyk, P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117, 2008.

Barto, A. G. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pp. 17–47. Springer, 2013.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018a.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018b.

Charikar, M. S. Similarity estimation techniques from rounding algorithms. In *ACM symposium on Theory of computing*, pp. 380–388. ACM, 2002.

Chentanez, N., Barto, A. G., and Singh, S. P. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 1281–1288, 2005.

Degris, T., White, M., and Sutton, R. S. Off-policy actor-critic. In *International Conference on Machine Learning*, 2012.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pp. 1407–1416. PMLR, 2018.

Fujimoto, S., Van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. 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.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lin, L.-J. *Reinforcement learning for robots using neural networks*. Carnegie Mellon University, 1992.

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

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. G. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.

Osband, I., Doron, Y., Hessel, M., Aslanides, J., Sezener, E., Saraiva, A., McKinney, K., Lattimore, T., Szepesvari, C., Singh, S., et al. Behaviour suite for reinforcement learning. In *International Conference on Learning Representations*, 2020.

Ostrovski, G., Bellemare, M. G., van den Oord, A., and Munos, R. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pp. 2721–2730. JMLR. org, 2017.

Oudeyer, P.-Y. and Kaplan, F. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2): 265–286, 2007.

Oudeyer, P.-Y., Kaplan, F., et al. How can we define intrinsic motivation. In *Conference on Epigenetic Robotics*, volume 5, pp. 29–31, 2008.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, volume 2017, 2017.

Raileanu, R. and Rocktäschel, T. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *International Conference on Learning Representations*, 2020.

Rashid, T., Peng, B., Böhmer, W., and Whiteson, S. Optimistic exploration even with a pessimistic initialisation. In *International Conference on Learning Representations*, 2020.

Schmidhuber, J. Curious model-building control systems. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, pp. 1458–1463. IEEE, 1991.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. PMLR, 2014.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Taïga, A. A., Fedus, W., Machado, M. C., Courville, A., and Bellemare, M. G. Benchmarking bonus-based exploration methods on the arcade learning environment. *arXiv preprint arXiv:1908.02388*, 2019.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2094–2100. AAAI Press, 2016.

Watkins, C. J. C. H. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

Whitney, W. F., Bloesch, M., Springenberg, J. T., Abdolmaleki, A., and Riedmiller, M. Rethinking exploration for sample-efficient policy learning. *arXiv preprint arXiv:2101.09458*, 2021.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

A. Hyperparameter Configuration

A gridsearch was conducted to identify the best hyperparameter configuration for each algorithm in both environments. In order to keep the search feasible, hyperparameter search for DeepSea is conducted in the task with $N = 20$ and the same configuration is applied in all other DeepSea tasks. Similarly, for the Hallway environment hyperparameter search has been conducted in the task with $N_l = N_r = 10$. Every gridsearch configuration is evaluated using 3 different random seeds with results being averaged over all seeds.

A.1. Baseline Hyperparameters

First, the best hyperparameters for A2C and PPO baselines were identified by training each configuration in the respective training task of both environments with Dict-Count intrinsic rewards. We chose to conduct the hyperparameter search with intrinsic rewards as both baselines do perform poorly in most tasks without any intrinsic rewards which make different hyperparameter configurations hardly distinguishable. The Dict-Count intrinsic reward definition was chosen as it does not require any hyperparameter tuning in contrast to other intrinsic reward definitions. For all considered hyperparameter combinations, we conduct training in the same way as described in Section 4 with $\lambda = 1$. The best hyperparameter configuration is chosen as the one which leads to highest maximum evaluation returns. If multiple configurations reach the same maximum evaluation returns then the one with the highest average evaluation returns computed over all 100 evaluations throughout training is considered the best. Such latter metric considers achieved returns alongside learning speed. Identified hyperparameters for A2C and PPO in both environments can be found in Table 2 and Table 3. All considered values of hyperparameters are listed with the best-identified combination highlighted in bold.

Table 2: Hyperparameters for A2C baseline.

Hyperparameter	DeepSea	Hallway
Normalise observations	False, **True**	**False**, True
Normalise rewards	False, **True**	**False**, True
Learning rate	$3e^{-4}$, $\mathbf{1e^{-3}}$	$\mathbf{3e^{-4}}$, $1e^{-3}$
Nonlinearity	Tanh, **ReLU**	**Tanh**, ReLU
Maximum gradient norm	$\mathbf{0.5}$, 10.0, 40.0	$\mathbf{0.5}$, 10.0, 40.0
Entropy loss coefficient	$\mathbf{1e^{-4}}$, $3e^{-4}$, $7e^{-4}$, $1e^{-3}$	$\mathbf{1e^{-4}}$, $3e^{-4}$, $7e^{-4}$, $1e^{-3}$
Actor architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
Critic architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
N-steps	**5**	**5**
Adam ϵ	**0.001**	**0.001**
Value loss coefficient	**0.5**	**0.5**

Table 3: Hyperparameters for PPO baseline.

Hyperparameter	DeepSea	Hallway
Normalise observations	**False**, True	**False**, True
Normalise rewards	**False**, True	**False**, True
Learning rate	$3e^{-4}$, $\mathbf{1e^{-3}}$	$\mathbf{3e^{-4}}$, $1e^{-3}$
Nonlinearity	**Tanh**, ReLU	Tanh, **ReLU**
Maximum gradient norm	$\mathbf{0.5}$, 10.0, 40.0	$\mathbf{0.5}$, 10.0, 40.0
Entropy loss coefficient	$\mathbf{1e^{-4}}$, $3e^{-4}$, $7e^{-4}$, $1e^{-3}$	$1e^{-4}$, $3e^{-4}$, $\mathbf{7e^{-4}}$, $1e^{-3}$
Actor architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
Critic architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
N-steps	**10**	**10**
Adam ϵ	**0.001**	**0.001**
Value loss coefficient	**0.5**	**0.5**
Number of epochs	**10**	**10**
Number of minibatches	**4**	**4**
Clipping ratio	**0.1**	**0.1**
Clip value loss	**True**	**True**

A.2. Intrinsic Reward Hyperparameters

Following the hyperparameter search of A2C and PPO with Dict-Count intrinsic rewards, a search over hyperparameters of all parameterised intrinsic reward definitions was conducted. The setup of the hyperparameter search is identical to the one described above and the best identified hyperparameter configuration for the baseline algorithms are used in the gridsearch for intrinsic rewards. Best identified hyperparameters and all considered hyperparameter configurations can be found in Tables 4 to 7.

Table 4: Hyperparameters for Hash-Count.

	Hyperparameter	DeepSea	Hallway
A2C	Hash key dimensionality	**16**, 32, 64, 128	**16**, 32, 64, 128
PPO	Hash key dimensionality	**16**, 32, 64, 128	**16**, 32, 64, 128

Table 5: Hyperparameters for ICM.

	Hyperparameter	DeepSea	Hallway
General	ϕ architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
	$\phi(s)$ dimensionality	**16**	**16**
	Forward prediction architecture	**Fully connected (64)**	**Fully connected (64)**
	Inverse prediction architecture	**Fully connected (64)**	**Fully connected (64)**
A2C	Learning rate	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$	$1e^{-7}, 5e^{-7}, \mathbf{1e^{-6}}, 5e^{-6}, 1e^{-5}$
	Forward loss coefficient	$0.5, 1.0, \mathbf{5.0}, 10.0$	$0.5, 1.0, \mathbf{5.0}, 10.0$
	Inverse loss coefficient	$0.5, \mathbf{1.0}, 5.0, 10.0$	$\mathbf{0.5}, 1.0, 5.0, 10.0$
PPO	Learning rate	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$
	Forward loss coefficient	$0.5, 1.0, \mathbf{5.0}, 10.0$	$\mathbf{0.5}, 1.0, 5.0, 10.0$
	Inverse loss coefficient	$0.5, \mathbf{1.0}, 5.0, 10.0$	$0.5, 1.0, 5.0, \mathbf{10.0}$

Table 6: Hyperparameters for RND.

	Hyperparameter	DeepSea	Hallway
General	ϕ architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
	$\phi(s)$ dimensionality	**16**	**16**
A2C	Learning rate	$\mathbf{1e^{-7}}, 5e^{-7}, 1e^{-6}, 5e^{-6}, 1e^{-5}$	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$
PPO	Learning rate	$\mathbf{1e^{-7}}, 5e^{-7}, 1e^{-6}, 5e^{-6}, 1e^{-5}$	$1e^{-7}, \mathbf{5e^{-7}}, 1e^{-6}, 5e^{-6}, 1e^{-5}$

Table 7: Hyperparameters for RIDE.

	Hyperparameter	DeepSea	Hallway
General	ϕ architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
	$\phi(s)$ dimensionality	**16**	**16**
	Forward prediction architecture	**Fully connected (64)**	**Fully connected (64)**
	Inverse prediction architecture	**Fully connected (64)**	**Fully connected (64)**
	State count	**Dict-Count**	**Dict-Count**
A2C	Learning rate	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$
	Forward loss coefficient	$\mathbf{0.5}, 1.0, 5.0, 10.0$	$0.5, 1.0, 5.0, \mathbf{10.0}$
	Inverse loss coefficient	$0.5, 1.0, 5.0, \mathbf{10.0}$	$\mathbf{0.5}, 1.0, 5.0, 10.0$
PPO	Learning rate	$1e^{-7}, 5e^{-7}, 1e^{-6}, \mathbf{5e^{-6}}, 1e^{-5}$	$\mathbf{1e^{-7}}, 5e^{-7}, 1e^{-6}, 5e^{-6}, 1e^{-5}$
	Forward loss coefficient	$0.5, 1.0, 5.0, \mathbf{10.0}$	$0.5, \mathbf{1.0}, 5.0, 10.0$
	Inverse loss coefficient	$0.5, \mathbf{1.0}, 5.0, 10.0$	$0.5, \mathbf{1.0}, 5.0, 10.0$

A.3. Decoupled Reinforcement Learning Hyperparameters

Based on the aforementioned hyperparameters, a gridsearch for all DeRL algorithms, DeA2C, DePPO and DeDQN, is conducted using the best identified A2C and Dict-Count intrinsic rewards to train the exploration policy. Identified configurations are listed in Tables 8 to 10. Table 11 shows hyperparameters for DeRL algorithms with ICM intrinsic rewards used to train the A2C exploration policy. The same general architecture is used for ICM when applied alongside DeRL as reported in Table 5.

Table 8: Hyperparameters for DeA2C.

Hyperparameter	DeepSea	Hallway
Importance sampling	**Default IS weights**, Retrace(λ)	Default IS weights, **Retrace(λ)**
Learning rate	$3e^{-4}$, $\mathbf{1e^{-3}}$	$\mathbf{3e^{-4}}$, $1e^{-3}$
Nonlinearity	Tanh, **ReLU**	**Tanh**, ReLU
Maximum gradient norm	$\mathbf{0.5}$	$\mathbf{0.5}$
Entropy loss coefficient	0.0, $\mathbf{1e^{-6}}$, $1e^{-5}$, $1e^{-4}$	0.0, $1e^{-6}$, $\mathbf{1e^{-5}}$, $1e^{-4}$
Actor architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
Critic architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
N-steps	**5**	**5**
Adam ϵ	**0.001**	**0.001**
Value loss coefficient	**0.5**	**0.5**
T_{Dec}	**1**	**1**

Table 9: Hyperparameters for DePPO.

Hyperparameter	DeepSea	Hallway
Importance sampling	**Default IS weights**	**Default IS weights**
Learning rate	$3e^{-4}$, $\mathbf{1e^{-3}}$	$\mathbf{3e^{-4}}$, $1e^{-3}$
Nonlinearity	Tanh, **ReLU**	Tanh, **ReLU**
Maximum gradient norm	$\mathbf{0.5}$	$\mathbf{0.5}$
Entropy loss coefficient	0.0, $1e^{-6}$, $1e^{-5}$, $\mathbf{1e^{-4}}$	0.0, $\mathbf{1e^{-6}}$, $1e^{-5}$, $1e^{-4}$
Actor architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
Critic architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
N-steps	**10**	**10**
Adam ϵ	**0.001**	**0.001**
Value loss coefficient	**0.5**	**0.5**
Number of epochs	**10**	**10**
Number of minibatches	**4**	**4**
Clipping ratio	**0.1**	**0.1**
Clip value loss	**True**	**True**
T_{Dec}	**1**	**1**

Table 10: Hyperparameters for DeDQN.

Hyperparameter	DeepSea	Hallway
Learning rate	$1e^{-4}$, $3e^{-4}$, $\mathbf{1e^{-3}}$	$\mathbf{1e^{-4}}$, $3e^{-4}$, $1e^{-3}$
Soft target update τ	**0.01**, 0.001	0.01, **0.001**
Batch size	128, **256**, 512	128, 256, **512**
N-steps	**5**	**5**
Nonlinearity	**Tanh**, ReLU	Tanh, **ReLU**
Replay buffer	**Default**, Prioritised	**Default**, Prioritised
Replay buffer capacity	**100,000**	**100,000**
Maximum gradient norm	$\mathbf{0.5}$	$\mathbf{0.5}$
DQN architecture	**Fully connected (64, 64)**	**Fully connected (64, 64)**
N-steps	**5**	**5**
Adam ϵ	**0.001**	**0.001**
T_{Dec}	**1**	**1**

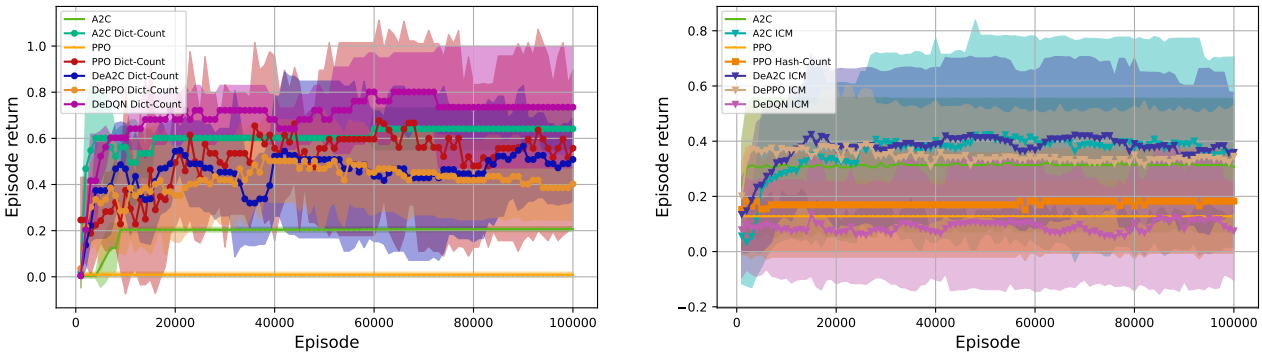
Table 11: Hyperparameters for DeRL with ICM.

	Hyperparameter	DeepSea	Hallway
DeA2C	Learning rate	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$	$1e^{-7}, 5e^{-7}, \mathbf{1e^{-6}}, 5e^{-6}, 1e^{-5}$
	Forward loss coefficient	$0.5, 1.0, \mathbf{5.0}, 10.0$	$0.5, 1.0, \mathbf{5.0}, 10.0$
	Inverse loss coefficient	$0.5, \mathbf{1.0}, 5.0, 10.0$	$\mathbf{0.5}, 1.0, 5.0, 10.0$
DePPO	Learning rate	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$
	Forward loss coefficient	$0.5, 1.0, \mathbf{5.0}, 10.0$	$\mathbf{0.5}, 1.0, 5.0, 10.0$
	Inverse loss coefficient	$0.5, \mathbf{1.0}, 5.0, 10.0$	$0.5, 1.0, 5.0, \mathbf{10.0}$
DeDQN	Learning rate	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$	$1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, \mathbf{1e^{-5}}$
	Forward loss coefficient	$0.5, 1.0, \mathbf{5.0}, 10.0$	$\mathbf{0.5}, 1.0, 5.0, 10.0$
	Inverse loss coefficient	$0.5, \mathbf{1.0}, 5.0, 10.0$	$0.5, 1.0, 5.0, \mathbf{10.0}$

B. Evaluation Results

In this section, we provide tables containing maximum evaluation returns for all baselines and DeRL algorithms in every task. As described in Section 4, results are averaged across five random seeds for the best identified hyperparameter configuration as reported in Appendix A. For maximum evaluation returns, we identify the single evaluation out of all 100 conducted evaluations with the maximum evaluation returns averaged across all seeds and report its value with the standard deviation across all seeds. Within tables, the highest performing algorithms for each task are highlighted in bold together with every algorithm within a single standard deviation of the highest return. Besides these tables, we also provide learning curves for all baselines and DeRL algorithms in every evaluated task.

Normalised evaluation returns of the highest performing baselines and DeRL algorithms in both environments are shown in Figure 8. For each task of an environment, we normalise returns to be within [0, 1] with 0 and 1 corresponding to the minimum and maximum achievable returns within the task, respectively.



(a) DeepSea

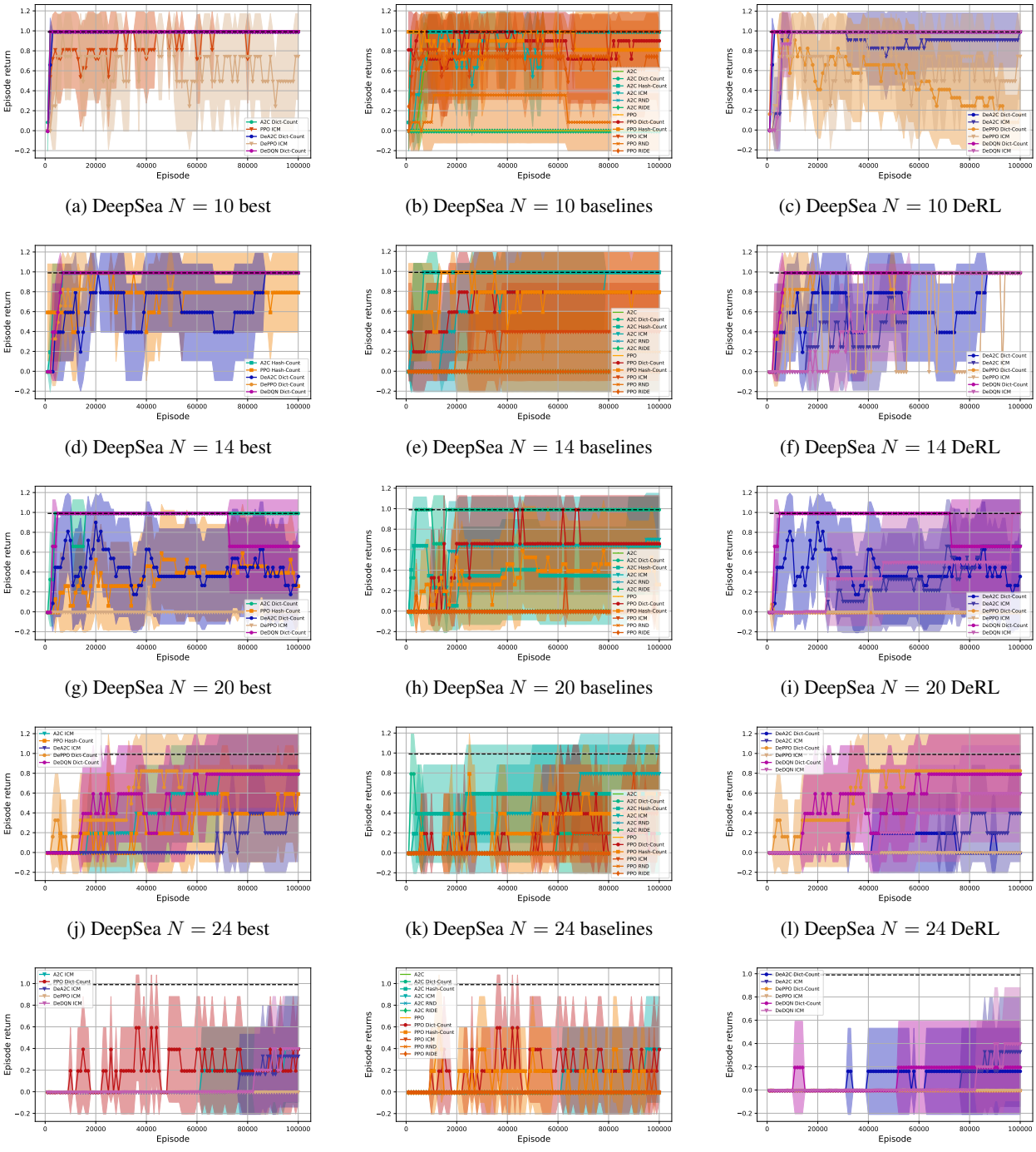
(b) Hallway

Figure 8: Normalised evaluation returns for (a) DeepSea and (b) Hallway. Returns for each task are normalised to be within [0, 1] before returns and standard deviations are averaged across all tasks of both respective environments.

B.1. DeepSea

Table 12: Maximum evaluation returns in the DeepSea environment.

Algorithm \ Task	DeepSea 10	DeepSea 14	DeepSea 20	DeepSea 24	DeepSea 30
A2C	$\mathbf{0.99 \pm 0.00}$	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
A2C Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.79 \pm 0.40}$	-0.01 ± 0.00
A2C Hash-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.64 ± 0.48	$\mathbf{0.59 \pm 0.49}$	0.00 ± 0.00
A2C ICM	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.70 ± 0.45	$\mathbf{0.79 \pm 0.40}$	$\mathbf{0.39 \pm 0.49}$
A2C RND	-0.01 ± 0.00	0.19 ± 0.40	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
A2C RIDE	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO Dict-Count	$\mathbf{0.99 \pm 0.00}$	0.79 ± 0.40	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.59 \pm 0.49}$	$\mathbf{0.59 \pm 0.49}$
PPO Hash-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.59 ± 0.49	$\mathbf{0.79 \pm 0.40}$	$\mathbf{0.59 \pm 0.49}$
PPO ICM	$\mathbf{0.99 \pm 0.00}$	0.39 ± 0.49	0.31 ± 0.46	$\mathbf{0.79 \pm 0.40}$	$\mathbf{0.19 \pm 0.40}$
PPO RND	0.36 ± 0.48	0.19 ± 0.40	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO RIDE	0.74 ± 0.43	0.00 ± 0.00	0.15 ± 0.36	0.00 ± 0.00	0.00 ± 0.00
DeA2C Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.90 ± 0.28	0.19 ± 0.40	$\mathbf{0.16 \pm 0.37}$
DeA2C ICM	$\mathbf{0.99 \pm 0.00}$	0.74 ± 0.43	0.66 ± 0.47	0.39 ± 0.49	$\mathbf{0.33 \pm 0.47}$
DePPO Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.08 ± 0.28	$\mathbf{0.82 \pm 0.37}$	0.00 ± 0.00
DePPO ICM	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.24 ± 0.43	0.00 ± 0.00	0.00 ± 0.00
DeDQN Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.79 \pm 0.40}$	$\mathbf{0.19 \pm 0.40}$
DeDQN ICM	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.59 \pm 0.49}$	$\mathbf{0.39 \pm 0.49}$



(a) DeepSea $N = 10$ best (b) DeepSea $N = 10$ baselines (c) DeepSea $N = 10$ DeRL

(d) DeepSea $N = 14$ best (e) DeepSea $N = 14$ baselines (f) DeepSea $N = 14$ DeRL

(g) DeepSea $N = 20$ best (h) DeepSea $N = 20$ baselines (i) DeepSea $N = 20$ DeRL

(j) DeepSea $N = 24$ best (k) DeepSea $N = 24$ baselines (l) DeepSea $N = 24$ DeRL

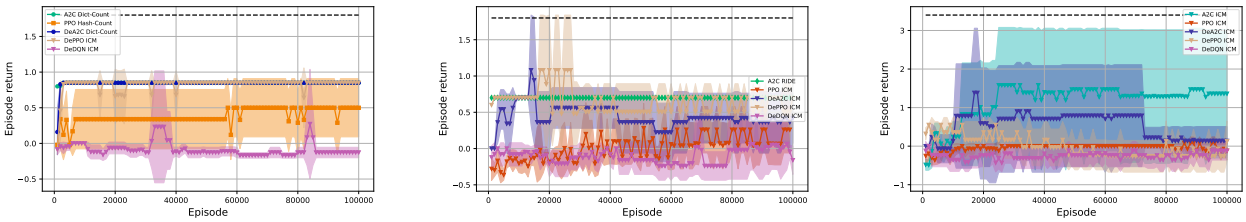
(m) DeepSea $N = 30$ best (n) DeepSea $N = 30$ baselines (o) DeepSea $N = 30$ DeRL

Figure 9: DeepSea evaluation returns for A2C and PPO with the highest achieving intrinsic reward and DeRL (first column), all baselines (second column) and all DeRL algorithms (third column) for all DeepSea tasks. Shading indicates a single standard deviation across all five seeds.

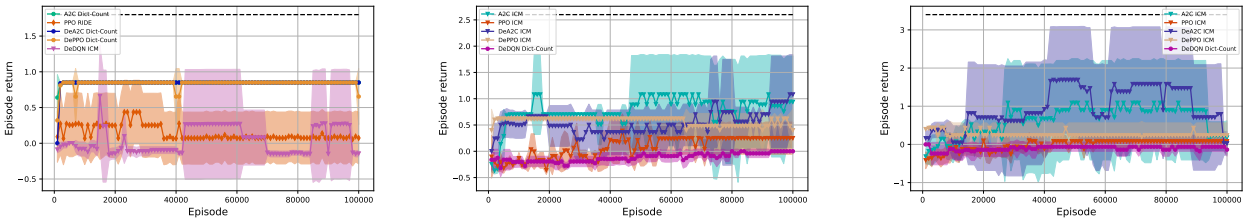
B.2. Hallway

Table 13: Maximum evaluation returns in the Hallway environment.

Algorithm \ Task	Hallway 10-0	Hallway 10-10	Hallway 20-0	Hallway 20-20	Hallway 30-0	Hallway 30-30
A2C	0.68 ± 0.34	0.51 ± 0.42	$\mathbf{0.52 \pm 0.30}$	$\mathbf{0.50 \pm 0.25}$	$\mathbf{0.44 \pm 0.22}$	$\mathbf{0.46 \pm 0.07}$
A2C Dict-Count	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.56 \pm 0.28}$	$\mathbf{0.60 \pm 0.00}$	$\mathbf{0.52 \pm 0.06}$	0.12 ± 0.41
A2C Hash-Count	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.56 \pm 0.28}$	$\mathbf{0.60 \pm 0.00}$	$\mathbf{0.52 \pm 0.06}$	0.22 ± 0.27
A2C ICM	0.68 ± 0.34	0.68 ± 0.34	$\mathbf{0.66 \pm 1.01}$	$\mathbf{1.08 \pm 0.76}$	$\mathbf{1.58 \pm 1.50}$	$\mathbf{1.09 \pm 1.15}$
A2C RND	0.12 ± 0.35	-0.06 ± 0.08	-0.17 ± 0.15	-0.24 ± 0.20	-0.24 ± 0.28	-0.12 ± 0.24
A2C RIDE	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.70 \pm 0.00}$	$\mathbf{0.62 \pm 0.04}$	$\mathbf{0.55 \pm 0.00}$	$\mathbf{0.43 \pm 0.06}$
PPO	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO Dict-Count	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO Hash-Count	0.50 ± 0.41	0.33 ± 0.41	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
PPO ICM	0.48 ± 0.39	0.32 ± 0.39	0.28 ± 0.34	$\mathbf{0.39 \pm 0.32}$	$\mathbf{0.08 \pm 0.16}$	0.10 ± 0.20
PPO RND	0.13 ± 0.37	0.24 ± 0.46	0.02 ± 0.31	0.17 ± 0.40	-0.04 ± 0.32	0.00 ± 0.00
PPO RIDE	0.10 ± 0.35	0.44 ± 0.45	-0.03 ± 0.37	0.13 ± 0.41	-0.19 ± 0.28	-0.01 ± 0.40
DeA2C Dict-Count	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.60 ± 0.00	$\mathbf{0.70 \pm 0.00}$	$\mathbf{0.55 \pm 0.00}$	$\mathbf{0.43 \pm 0.06}$
DeA2C ICM	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{1.08 \pm 0.76}$	$\mathbf{1.08 \pm 0.76}$	$\mathbf{1.38 \pm 1.68}$	$\mathbf{1.69 \pm 1.40}$
DePPO Dict-Count	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.56 ± 0.28	$\mathbf{0.60 \pm 0.00}$	$\mathbf{0.52 \pm 0.06}$	$\mathbf{0.43 \pm 0.06}$
DePPO ICM	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{1.08 \pm 0.76}$	$\mathbf{0.62 \pm 0.04}$	$\mathbf{0.55 \pm 0.00}$	$\mathbf{0.43 \pm 0.06}$
DeDQN Dict-Count	0.13 ± 0.37	0.13 ± 0.37	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
DeDQN ICM	0.27 ± 0.77	0.66 ± 0.71	0.06 ± 0.36	$\mathbf{0.35 \pm 1.12}$	0.00 ± 0.00	0.00 ± 0.00

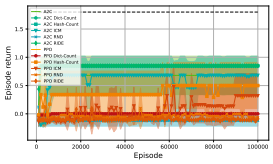


(a) Hallway $N_l = 10, N_r = 0$ (b) Hallway $N_l = 20, N_r = 0$ (c) Hallway $N_l = 30, N_r = 0$

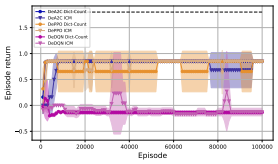


(d) Hallway $N_l = 10, N_r = 10$ (e) Hallway $N_l = 20, N_r = 20$ (f) Hallway $N_l = 30, N_r = 30$

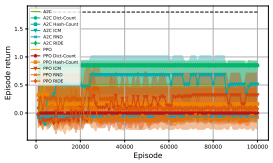
Figure 10: Hallway evaluation returns for A2C and PPO with the highest achieving intrinsic reward and DeRL for all Hallway tasks. Shading indicates a single standard deviation across all five seeds.



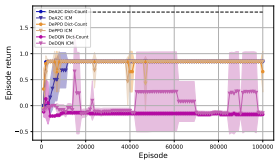
(a) Hallway $(10, 0)$ baselines



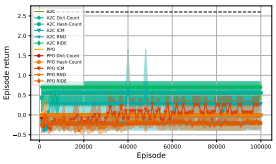
(b) Hallway $(10, 0)$ DeRL



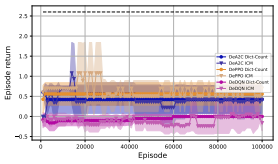
(c) Hallway $(10, 10)$ baselines



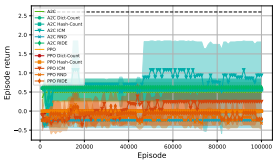
(d) Hallway $(10, 10)$ DeRL



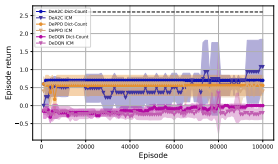
(e) Hallway $(20, 0)$ baselines



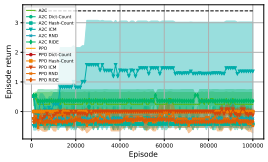
(f) Hallway $(20, 0)$ DeRL



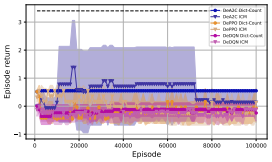
(g) Hallway $(20, 20)$ baselines



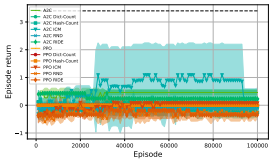
(h) Hallway $(20, 20)$ DeRL



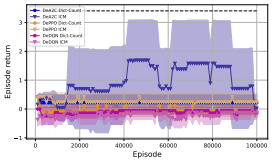
(i) Hallway $(30, 0)$ baselines



(j) Hallway $(30, 0)$ DeRL



(k) Hallway $(30, 30)$ baselines



(l) Hallway $(30, 30)$ DeRL

Figure 11: Hallway evaluation returns for all baselines and DeRL algorithms for all Hallway tasks. Shading indicates a single standard deviation across all five seeds.

C. Hyperparameter Sensitivity

In this section, we provide tables containing maximum and average achieved evaluation returns for both sets of hyerparameter sensitivity experiments. As described in Section 6, we evaluate both baselines and DeRL algorithms with varying intrinsic reward coefficients, λ, and speed of decay of intrinsic rewards. Both sets of experiments are done in the DeepSea 10 and the Hallway task with $N_l = N_r = 10$. We provide tables showing the maximum and average evaluation returns as in Appendix B and bar plots indicating the varying average evaluation returns for various parameterisation of intrinsic rewards. Within tables, the highest performing configuration for a single algorithm is highlighted in bold within each row with all configurations within a single standard deviation of the highest return.

C.1. Intrinsic Reward Scale

First, we evaluate all baselines and DeRL algorithms for various intrinsic reward coefficients $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$ in DeepSea and Hallway.

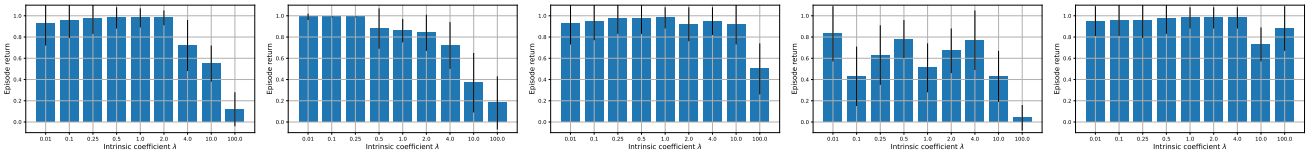
C.1.1. DeepSea

Table 14: Maximum evaluation returns in DeepSea 10 with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.

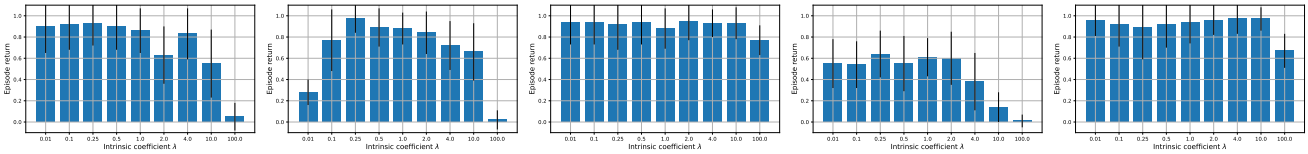
Algorithm \ λ	0.01	0.1	0.25	0.5	1.0	2.0	4.0	10.0	100.0
A2C Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.66 ± 0.47	0.33 ± 0.47
A2C Hash-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.66 ± 0.47
A2C ICM	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.66 ± 0.47
A2C RND	$\mathbf{0.99 \pm 0.00}$	0.33 ± 0.47	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	0.00 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00
A2C RIDE	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
PPO Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
PPO Hash-Count	0.66 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.66 ± 0.47
PPO ICM	0.33 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.33 ± 0.47
PPO RND	0.33 ± 0.47	0.00 ± 0.00	0.33 ± 0.47	-0.01 ± 0.00	$\mathbf{0.36 \pm 0.48}$	0.00 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00
PPO RIDE	0.33 ± 0.47	0.66 ± 0.47	0.66 ± 0.47	0.66 ± 0.47	$\mathbf{0.74 \pm 0.43}$	0.66 ± 0.47	0.33 ± 0.47	0.66 ± 0.47	0.33 ± 0.47
DeA2C Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DeA2C ICM	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DePPO Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.74 ± 0.43	0.49 ± 0.50
DePPO ICM	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.49 ± 0.50	0.49 ± 0.50
DeDQN Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DeDQN ICM	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$

Table 15: Average evaluation returns in DeepSea 10 with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.

Algorithm \ λ	0.01	0.1	0.25	0.5	1.0	2.0	4.0	10.0	100.0
A2C Dict-Count	0.93 ± 0.21	0.96 ± 0.17	0.97 ± 0.14	$\mathbf{0.98 \pm 0.10}$	$\mathbf{0.98 \pm 0.09}$	$\mathbf{0.98 \pm 0.07}$	0.72 ± 0.24	0.55 ± 0.17	0.12 ± 0.16
A2C Hash-Count	0.94 ± 0.21	0.96 ± 0.17	0.97 ± 0.14	$\mathbf{0.98 \pm 0.10}$	$\mathbf{0.98 \pm 0.09}$	$\mathbf{0.98 \pm 0.07}$	0.88 ± 0.19	0.31 ± 0.15	0.11 ± 0.22
A2C ICM	0.90 ± 0.25	0.92 ± 0.24	$\mathbf{0.93 \pm 0.21}$	0.90 ± 0.22	0.86 ± 0.21	0.63 ± 0.27	0.83 ± 0.24	0.55 ± 0.32	0.05 ± 0.13
A2C RND	$\mathbf{0.91 \pm 0.24}$	0.31 ± 0.08	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00
A2C RIDE	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
PPO Dict-Count	$\mathbf{0.99 \pm 0.03}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.88 ± 0.19	0.86 ± 0.11	0.84 ± 0.17	0.72 ± 0.22	0.37 ± 0.28	0.18 ± 0.25
PPO Hash-Count	0.66 ± 0.03	$\mathbf{0.99 \pm 0.00}$	0.98 ± 0.05	0.96 ± 0.10	0.87 ± 0.08	0.98 ± 0.07	0.82 ± 0.26	0.55 ± 0.24	0.15 ± 0.22
PPO ICM	0.28 ± 0.12	0.77 ± 0.28	$\mathbf{0.97 \pm 0.13}$	0.89 ± 0.18	0.88 ± 0.15	0.84 ± 0.20	0.72 ± 0.23	0.66 ± 0.27	0.02 ± 0.09
PPO RND	$\mathbf{0.33 \pm 0.00}$	-0.01 ± 0.00	0.01 ± 0.07	-0.01 ± 0.00	0.23 ± 0.14	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00
PPO RIDE	0.29 ± 0.11	0.63 ± 0.10	0.64 ± 0.07	0.59 ± 0.14	$\mathbf{0.70 \pm 0.09}$	0.64 ± 0.07	0.29 ± 0.11	0.64 ± 0.07	0.32 ± 0.03
DeA2C Dict-Count	0.93 ± 0.20	0.95 ± 0.18	0.97 ± 0.14	0.97 ± 0.14	$\mathbf{0.98 \pm 0.10}$	0.92 ± 0.16	0.95 ± 0.13	0.92 ± 0.19	0.50 ± 0.24
DeA2C ICM	0.94 ± 0.21	0.94 ± 0.21	0.92 ± 0.24	0.94 ± 0.21	0.88 ± 0.19	$\mathbf{0.95 \pm 0.18}$	0.93 ± 0.13	0.93 ± 0.15	0.77 ± 0.14
DePPO Dict-Count	$\mathbf{0.83 \pm 0.26}$	0.43 ± 0.28	0.63 ± 0.28	0.78 ± 0.18	0.51 ± 0.23	0.67 ± 0.21	0.77 ± 0.28	0.43 ± 0.24	0.04 ± 0.12
DePPO ICM	0.55 ± 0.23	0.54 ± 0.22	$\mathbf{0.64 \pm 0.22}$	0.55 ± 0.26	0.61 ± 0.18	0.60 ± 0.25	0.38 ± 0.27	0.14 ± 0.14	0.01 ± 0.06
DeDQN Dict-Count	0.95 ± 0.14	0.96 ± 0.15	0.96 ± 0.17	0.97 ± 0.14	$\mathbf{0.98 \pm 0.10}$	$\mathbf{0.98 \pm 0.10}$	$\mathbf{0.98 \pm 0.10}$	0.73 ± 0.16	0.88 ± 0.21
DeDQN ICM	0.96 ± 0.15	0.92 ± 0.21	0.89 ± 0.30	0.92 ± 0.22	0.94 ± 0.20	0.96 ± 0.14	$\mathbf{0.97 \pm 0.14}$	$\mathbf{0.97 \pm 0.11}$	0.67 ± 0.16

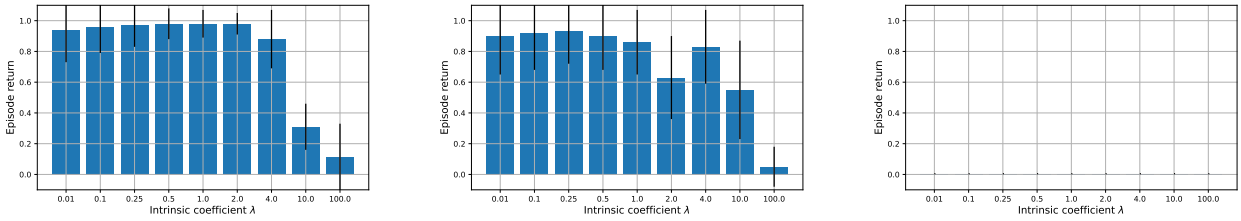


(a) A2C Dict-Count (b) PPO Dict-Count (c) DeA2C Dict-Count (d) DePPO Dict-Count (e) DeDQN Dict-Count

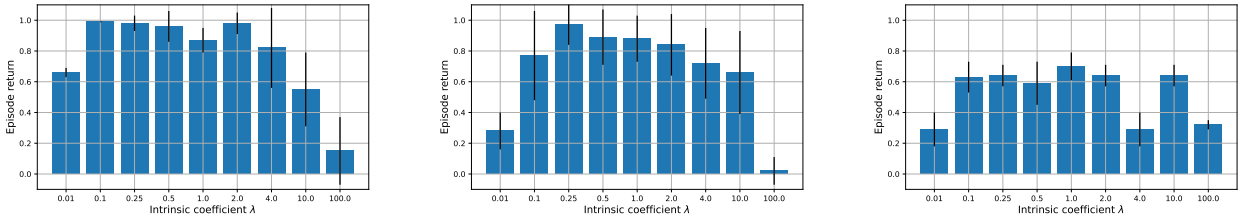


(f) A2C ICM (g) PPO ICM (h) DeA2C ICM (i) DePPO ICM (j) DeDQN ICM

Figure 12: Average evaluation returns for baselines and DeRL with Dict-Count and ICM intrinsic rewards in DeepSea 10 with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.



(a) A2C Hash-Count (b) A2C ICM (c) A2C RIDE



(d) PPO Hash-Count (e) PPO ICM (f) PPO RIDE

Figure 13: Average evaluation returns for A2C and PPO with Hash-Count, ICM and RIDE intrinsic rewards in DeepSea 10 with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.

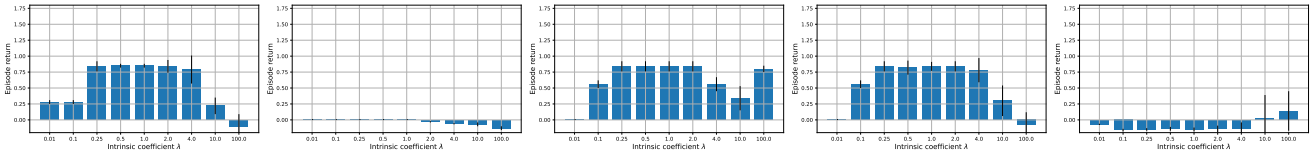
C.1.2. HALLWAY

Table 16: Maximum evaluation returns in Hallway $N_l = N_r = 10$ with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.

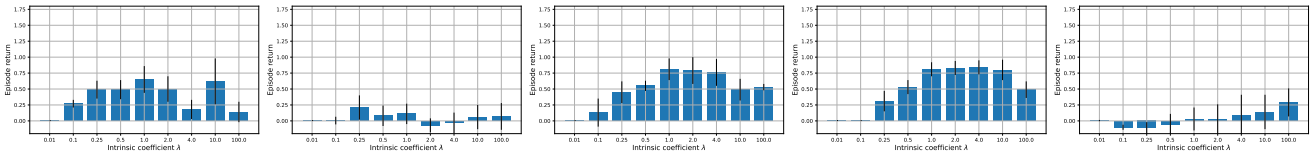
Algorithm \ λ	0.01	0.1	0.25	0.5	1.0	2.0	4.0	10.0	100.0
A2C Dict-Count	0.28 ± 0.40	0.28 ± 0.40	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.28 ± 0.40	0.80 ± 0.00
A2C Hash-Count	0.28 ± 0.40	0.28 ± 0.40	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.28 ± 0.40	0.80 ± 0.00
A2C ICM	0.00 ± 0.00	0.28 ± 0.40	0.85 ± 0.00	$\mathbf{1.17 \pm 0.45}$	0.77 ± 0.24	0.57 ± 0.40	0.55 ± 0.39	0.85 ± 0.00	0.25 ± 0.43
A2C RND	0.00 ± 0.00	$\mathbf{0.57 \pm 0.40}$	0.25 ± 0.43	-0.10 ± 0.08	-0.08 ± 0.08	-0.13 ± 0.09	-0.13 ± 0.09	-0.10 ± 0.08	-0.10 ± 0.08
A2C RIDE	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$
PPO Dict-Count	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	-0.03 ± 0.05	$\mathbf{0.00 \pm 0.00}$	-0.07 ± 0.09
PPO Hash-Count	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.30 ± 0.40	0.54 ± 0.38	$\mathbf{0.80 \pm 0.00}$	0.28 ± 0.39	0.27 ± 0.38
PPO ICM	0.00 ± 0.00	0.27 ± 0.38	0.55 ± 0.39	0.28 ± 0.40	0.43 ± 0.41	0.25 ± 0.43	0.47 ± 0.47	0.80 ± 0.00	$\mathbf{0.83 \pm 0.78}$
PPO RND	0.00 ± 0.00	0.54 ± 0.38	0.50 ± 0.43	0.27 ± 0.38	0.47 ± 0.44	0.54 ± 0.38	$\mathbf{0.80 \pm 0.00}$	0.54 ± 0.38	$\mathbf{0.80 \pm 0.00}$
PPO RIDE	0.23 ± 0.41	0.23 ± 0.41	0.23 ± 0.41	0.23 ± 0.41	$\mathbf{0.32 \pm 0.44}$	0.23 ± 0.41	0.23 ± 0.41	0.23 ± 0.41	0.23 ± 0.41
DeA2C Dict-Count	0.00 ± 0.00	0.57 ± 0.40	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.83 ± 0.02	0.80 ± 0.00	0.82 ± 0.02
DeA2C ICM	0.00 ± 0.00	0.57 ± 0.40	0.57 ± 0.40	0.57 ± 0.40	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.83 ± 0.02	0.57 ± 0.40	0.82 ± 0.02
DePPO Dict-Count	0.00 ± 0.00	0.57 ± 0.40	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.57 ± 0.40	0.80 ± 0.00
DePPO ICM	0.00 ± 0.00	0.00 ± 0.00	0.57 ± 0.40	0.57 ± 0.40	0.85 ± 0.00	0.85 ± 0.00	0.85 ± 0.00	$\mathbf{1.17 \pm 0.45}$	0.57 ± 0.40
DeDQN Dict-Count	-0.04 ± 0.06	-0.07 ± 0.09	-0.07 ± 0.09	0.00 ± 0.00	0.06 ± 0.26	0.22 ± 0.46	0.57 ± 0.40	$\mathbf{1.48 \pm 0.45}$	0.85 ± 0.00
DeDQN ICM	0.00 ± 0.00	0.00 ± 0.00	0.49 ± 0.92	0.52 ± 0.91	0.42 ± 0.85	0.56 ± 0.88	0.82 ± 0.02	0.60 ± 0.85	$\mathbf{0.83 \pm 0.02}$

Table 17: Average evaluation returns in Hallway $N_l = N_r = 10$ with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.

Algorithm \ λ	0.01	0.1	0.25	0.5	1.0	2.0	4.0	10.0	100.0
A2C Dict-Count	0.28 ± 0.03	0.28 ± 0.03	0.84 ± 0.08	$\mathbf{0.85 \pm 0.03}$	$\mathbf{0.85 \pm 0.03}$	0.84 ± 0.10	0.79 ± 0.22	0.22 ± 0.13	-0.10 ± 0.19
A2C Hash-Count	0.28 ± 0.03	0.28 ± 0.03	0.84 ± 0.08	$\mathbf{0.85 \pm 0.03}$	$\mathbf{0.85 \pm 0.03}$	0.84 ± 0.10	0.79 ± 0.21	0.01 ± 0.12	-0.10 ± 0.19
A2C ICM	0.00 ± 0.00	0.27 ± 0.06	0.49 ± 0.14	0.49 ± 0.15	$\mathbf{0.65 \pm 0.21}$	0.50 ± 0.20	0.18 ± 0.15	0.62 ± 0.36	0.14 ± 0.16
A2C RND	0.00 ± 0.00	$\mathbf{0.56 \pm 0.07}$	0.24 ± 0.03	-0.10 ± 0.00	-0.08 ± 0.00	-0.13 ± 0.00	-0.13 ± 0.00	-0.13 ± 0.01	-0.11 ± 0.02
A2C RIDE	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$
PPO Dict-Count	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	-0.03 ± 0.01	-0.06 ± 0.01	-0.07 ± 0.03	-0.13 ± 0.03
PPO Hash-Count	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.11 ± 0.07	0.28 ± 0.04	$\mathbf{0.29 \pm 0.12}$	0.05 ± 0.10	-0.02 ± 0.09
PPO ICM	0.00 ± 0.00	0.00 ± 0.06	$\mathbf{0.21 \pm 0.19}$	0.08 ± 0.16	0.11 ± 0.16	-0.07 ± 0.11	-0.03 ± 0.16	0.06 ± 0.19	0.07 ± 0.21
PPO RND	0.00 ± 0.00	0.01 ± 0.12	-0.03 ± 0.10	0.03 ± 0.17	-0.04 ± 0.11	-0.07 ± 0.13	0.11 ± 0.20	0.07 ± 0.23	$\mathbf{0.25 \pm 0.15}$
PPO RIDE	-0.01 ± 0.14	-0.01 ± 0.14	-0.01 ± 0.14	-0.01 ± 0.14	$\mathbf{0.06 \pm 0.11}$	-0.01 ± 0.14	-0.01 ± 0.14	-0.01 ± 0.14	-0.01 ± 0.14
DeA2C Dict-Count	0.00 ± 0.00	0.56 ± 0.06	$\mathbf{0.84 \pm 0.08}$	$\mathbf{0.84 \pm 0.08}$	$\mathbf{0.84 \pm 0.08}$	$\mathbf{0.84 \pm 0.08}$	0.56 ± 0.11	0.34 ± 0.19	0.80 ± 0.05
DeA2C ICM	0.00 ± 0.00	0.13 ± 0.22	0.45 ± 0.17	0.55 ± 0.08	$\mathbf{0.81 \pm 0.17}$	0.79 ± 0.21	0.76 ± 0.21	0.49 ± 0.17	0.53 ± 0.05
DePPO Dict-Count	0.00 ± 0.00	0.56 ± 0.06	$\mathbf{0.84 \pm 0.08}$	0.82 ± 0.11	$\mathbf{0.84 \pm 0.07}$	$\mathbf{0.84 \pm 0.08}$	0.78 ± 0.19	0.30 ± 0.24	-0.08 ± 0.20
DePPO ICM	0.00 ± 0.00	0.00 ± 0.00	0.31 ± 0.16	0.53 ± 0.11	0.81 ± 0.11	0.83 ± 0.11	$\mathbf{0.84 \pm 0.11}$	0.80 ± 0.16	0.49 ± 0.13
DeDQN Dict-Count	-0.07 ± 0.01	-0.16 ± 0.02	-0.15 ± 0.02	-0.14 ± 0.03	-0.15 ± 0.03	-0.14 ± 0.05	-0.13 ± 0.09	0.02 ± 0.37	$\mathbf{0.14 \pm 0.31}$
DeDQN ICM	0.00 ± 0.01	-0.10 ± 0.04	-0.11 ± 0.09	-0.06 ± 0.17	0.03 ± 0.18	0.03 ± 0.23	0.09 ± 0.32	0.14 ± 0.27	$\mathbf{0.29 \pm 0.22}$

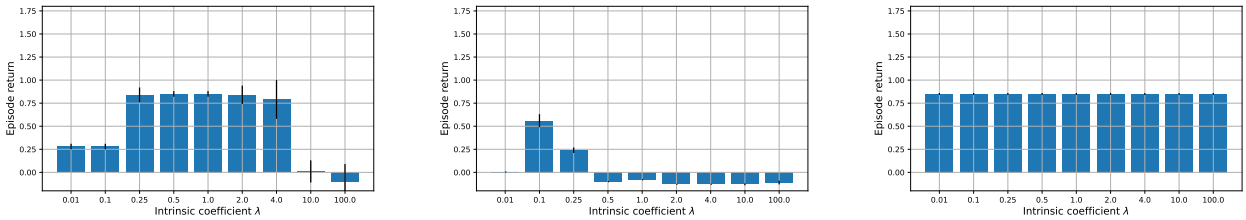


(a) A2C Dict-Count (b) PPO Dict-Count (c) DeA2C Dict-Count (d) DePPO Dict-Count (e) DeDQN Dict-Count

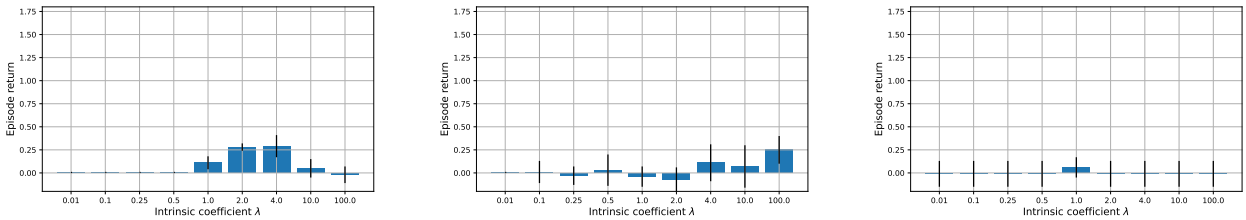


(f) A2C ICM (g) PPO ICM (h) DeA2C ICM (i) DePPO ICM (j) DeDQN ICM

Figure 14: Average evaluation returns for baselines and DeRL with Dict-Count and ICM intrinsic rewards in Hallway $N_l = N_r = 10$ with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.



(a) A2C Hash-Count (b) A2C RND (c) A2C RIDE



(d) PPO Hash-Count (e) PPO RND (f) PPO RIDE

Figure 15: Average evaluation returns for A2C and PPO with Hash-Count, RND and RIDE intrinsic rewards in Hallway $N_l = N_r = 10$ with $\lambda \in \{0.01, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 10.0, 100.0\}$.

C.2. Intrinsic Reward Decay

Second, we evaluate all baselines and DeRL algorithms with varying speed of decay of intrinsic rewards in DeepSea and Hallway. For count-based intrinsic rewards, the speed of decay can be determined by the increment of the state count $N(s)$. By default, the count is incremented by 1 for each state occurrence. For this analysis, we consider increments $\{0.01, 0.1, 0.2, 1.0, 5.0, 10.0, 100.0\}$. For deep prediction-based intrinsic rewards, ICM, RND and RIDE, the speed of decay is determined by their learning rates. We consider learning rates $\{1e^{-9}, 1e^{-8}, 2e^{-8}, 1e^{-7}, 5e^{-7}, 1e^{-6}, 1e^{-5}, 1e^{-4}, 1e^{-3}\}$.

C.2.1. DEEPSEA

Table 18: Maximum evaluation returns in DeepSea 10 with count-based intrinsic rewards and varying count increments.

Algorithm \ Count increment	0.01	0.1	0.2	1.0	5.0	10.0	100.0
A2C Dict-Count	0.66 ± 0.47	0.66 ± 0.47	0.66 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
A2C Hash-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
PPO Dict-Count	0.66 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.00 ± 0.00
PPO Hash-Count	0.66 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DeA2C Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DePPO Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DeDQN Dict-Count	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$

Table 19: Average evaluation returns in DeepSea 10 with count-based intrinsic rewards and varying count increments.

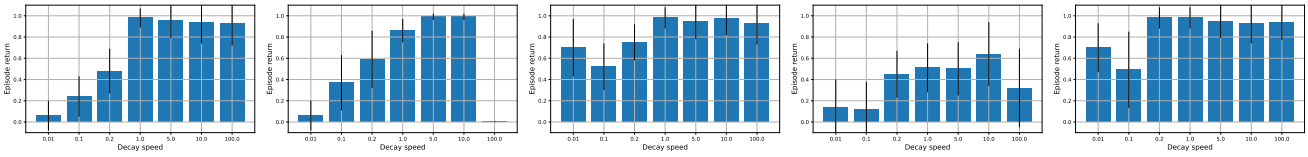
Algorithm \ Count increment	0.01	0.1	0.2	1.0	5.0	10.0	100.0
A2C Dict-Count	0.06 ± 0.14	0.24 ± 0.19	0.48 ± 0.21	$\mathbf{0.98 \pm 0.09}$	0.96 ± 0.17	0.94 ± 0.20	0.93 ± 0.21
A2C Hash-Count	0.72 ± 0.31	0.95 ± 0.14	$\mathbf{0.98 \pm 0.07}$	$\mathbf{0.98 \pm 0.09}$	0.97 ± 0.12	0.97 ± 0.14	0.96 ± 0.17
PPO Dict-Count	0.06 ± 0.14	0.37 ± 0.26	0.59 ± 0.27	0.86 ± 0.11	$\mathbf{0.99 \pm 0.03}$	$\mathbf{0.99 \pm 0.03}$	0.00 ± 0.00
PPO Hash-Count	0.40 ± 0.22	0.81 ± 0.23	0.73 ± 0.19	0.87 ± 0.08	0.78 ± 0.16	0.98 ± 0.07	$\mathbf{0.99 \pm 0.00}$
DeA2C Dict-Count	0.70 ± 0.27	0.52 ± 0.22	0.75 ± 0.17	$\mathbf{0.98 \pm 0.10}$	0.95 ± 0.17	0.97 ± 0.15	0.93 ± 0.20
DePPO Dict-Count	0.14 ± 0.26	0.12 ± 0.26	0.45 ± 0.22	0.51 ± 0.23	0.50 ± 0.25	$\mathbf{0.64 \pm 0.30}$	0.32 ± 0.37
DeDQN Dict-Count	0.70 ± 0.23	0.49 ± 0.36	$\mathbf{0.98 \pm 0.10}$	$\mathbf{0.98 \pm 0.10}$	0.95 ± 0.16	0.93 ± 0.19	0.94 ± 0.17

Table 20: Maximum evaluation returns in DeepSea 10 with prediction-based intrinsic rewards and varying learning rates.

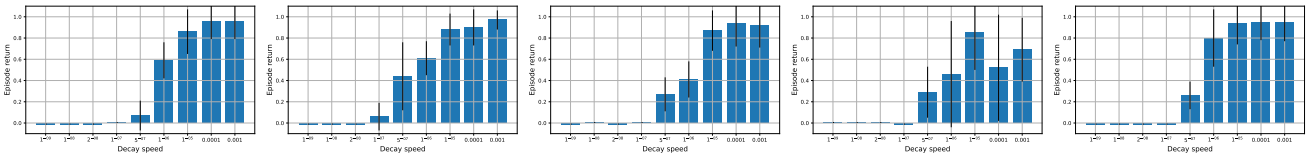
Algorithm \ Learning rate	1e-09	1e-08	2e-08	1e-07	5e-07	1e-06	1e-05	0.0001	0.001
A2C ICM	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	0.00 ± 0.00	0.33 ± 0.47	0.66 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
A2C RND	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	0.00 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	$\mathbf{0.33 \pm 0.47}$
A2C RIDE	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
PPO ICM	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	0.33 ± 0.47	$\mathbf{0.99 \pm 0.00}$	0.66 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
PPO RND	-0.01 ± 0.00	-0.01 ± 0.00	0.33 ± 0.47	$\mathbf{0.36 \pm 0.48}$	0.33 ± 0.47	0.33 ± 0.47	-0.01 ± 0.00	-0.01 ± 0.00	0.33 ± 0.47
PPO RIDE	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.33 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.66 ± 0.47	0.33 ± 0.47
DeA2C ICM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.66 ± 0.47	0.49 ± 0.50	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DePPO ICM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-0.01 ± 0.00	0.49 ± 0.50	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$
DeDQN ICM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.33 ± 0.47	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$

Table 21: Average evaluation returns in DeepSea 10 with prediction-based intrinsic rewards and varying learning rates.

Algorithm \ Learning rate	1e-09	1e-08	2e-08	1e-07	5e-07	1e-06	1e-05	0.0001	0.001
A2C ICM	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	0.00 ± 0.00	0.07 ± 0.14	0.59 ± 0.17	0.86 ± 0.21	$\mathbf{0.96 \pm 0.17}$	$\mathbf{0.96 \pm 0.17}$
A2C RND	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	$\mathbf{0.32 \pm 0.03}$
A2C RIDE	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
PPO ICM	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	0.06 ± 0.13	0.44 ± 0.32	0.61 ± 0.16	0.88 ± 0.15	0.90 ± 0.17	$\mathbf{0.97 \pm 0.09}$
PPO RND	-0.01 ± 0.00	-0.01 ± 0.00	$\mathbf{0.30 \pm 0.09}$	0.23 ± 0.14	0.18 ± 0.16	0.24 ± 0.15	-0.01 ± 0.00	-0.01 ± 0.00	$\mathbf{0.30 \pm 0.10}$
PPO RIDE	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-0.01 ± 0.00	0.17 ± 0.17	0.59 ± 0.27	$\mathbf{0.91 \pm 0.18}$	0.64 ± 0.08	0.30 ± 0.10
DeA2C ICM	-0.01 ± 0.00	0.00 ± 0.00	-0.01 ± 0.00	0.00 ± 0.00	0.27 ± 0.16	0.41 ± 0.17	0.87 ± 0.19	$\mathbf{0.94 \pm 0.22}$	0.92 ± 0.21
DePPO ICM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-0.01 ± 0.00	0.29 ± 0.24	0.46 ± 0.50	$\mathbf{0.85 \pm 0.35}$	0.52 ± 0.50	0.69 ± 0.30
DeDQN ICM	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	0.26 ± 0.13	0.80 ± 0.27	0.94 ± 0.20	$\mathbf{0.95 \pm 0.17}$	$\mathbf{0.95 \pm 0.18}$

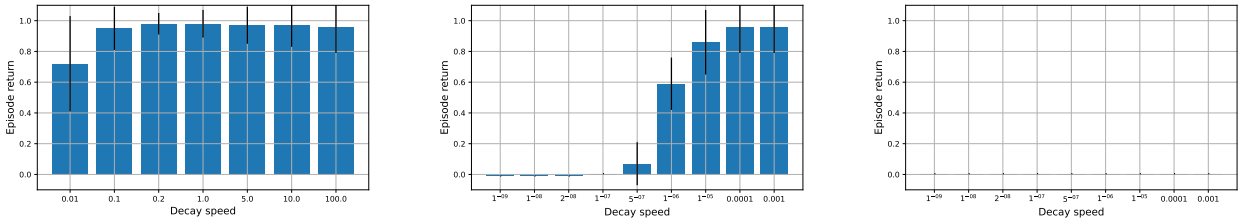


(a) A2C Dict-Count (b) PPO Dict-Count (c) DeA2C Dict-Count (d) DePPO Dict-Count (e) DeDQN Dict-Count

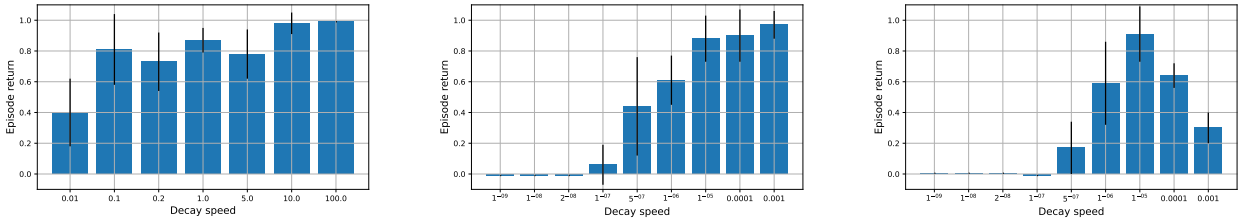


(f) A2C ICM (g) PPO ICM (h) DeA2C ICM (i) DePPO ICM (j) DeDQN ICM

Figure 16: Average evaluation returns for baselines and DeRL with Dict-Count and ICM intrinsic rewards in DeepSea 10 with varying count increments and learning rates.



(a) A2C Hash-Count (b) A2C ICM (c) A2C RIDE



(d) PPO Hash-Count (e) PPO ICM (f) PPO RIDE

Figure 17: Average evaluation returns for A2C and PPO with Hash-Count, ICM and RIDE intrinsic rewards in DeepSea 10 with varying count increments and learning rates.

C.2.2. HALLWAY

Table 22: Maximum evaluation returns in Hallway $N_l = N_r = 10$ with count-based intrinsic rewards and varying count increments.

Algorithm \ Count increment	0.01	0.1	0.2	1.0	5.0	10.0	100.0
A2C Dict-Count	$\mathbf{1.13 \pm 0.47}$	0.80 ± 0.00	0.00 ± 0.00	0.85 ± 0.00	0.28 ± 0.40	0.28 ± 0.40	0.28 ± 0.40
A2C Hash-Count	0.28 ± 0.40	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.28 ± 0.40
PPO Dict-Count	-0.03 ± 0.05	-0.03 ± 0.05	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
PPO Hash-Count	0.28 ± 0.40	$\mathbf{0.80 \pm 0.00}$	0.53 ± 0.38	0.30 ± 0.40	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
DeA2C Dict-Count	$\mathbf{0.85 \pm 0.00}$	0.82 ± 0.02	0.82 ± 0.02	$\mathbf{0.85 \pm 0.00}$	0.57 ± 0.40	0.00 ± 0.00	0.00 ± 0.00
DePPO Dict-Count	$\mathbf{1.48 \pm 0.45}$	0.80 ± 0.00	0.18 ± 0.44	0.85 ± 0.00	0.57 ± 0.40	0.57 ± 0.40	0.00 ± 0.00
DeDQN Dict-Count	0.85 ± 0.00	0.85 ± 0.00	$\mathbf{1.48 \pm 0.45}$	0.06 ± 0.26	0.20 ± 0.43	-0.04 ± 0.06	-0.07 ± 0.09

Table 23: Average evaluation returns in Hallway $N_l = N_r = 10$ with count-based intrinsic rewards and varying count increments.

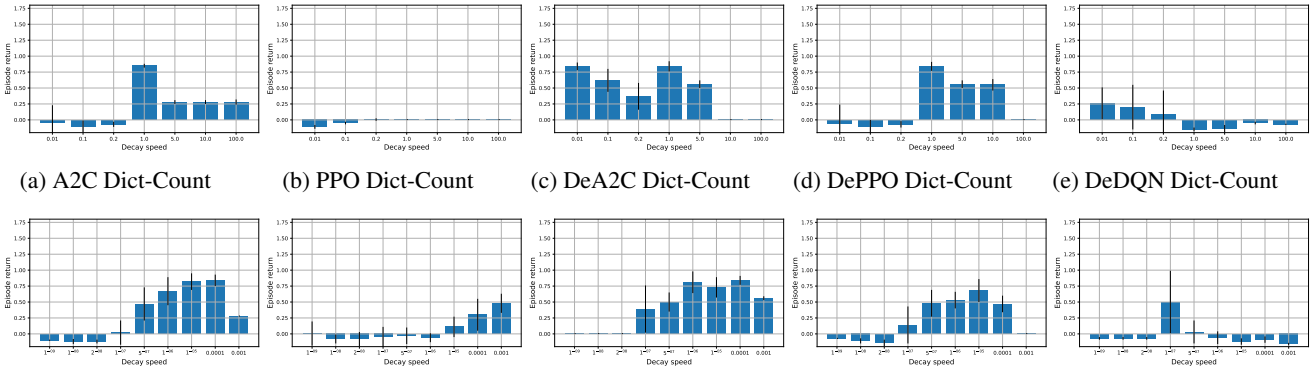
Algorithm \ Count increment	0.01	0.1	0.2	1.0	5.0	10.0	100.0
A2C Dict-Count	-0.04 ± 0.27	-0.10 ± 0.09	-0.07 ± 0.04	$\mathbf{0.85 \pm 0.03}$	0.28 ± 0.03	0.28 ± 0.03	0.28 ± 0.04
A2C Hash-Count	0.23 ± 0.13	0.82 ± 0.16	0.84 ± 0.10	$\mathbf{0.85 \pm 0.03}$	$\mathbf{0.85 \pm 0.03}$	0.84 ± 0.08	0.28 ± 0.03
PPO Dict-Count	-0.11 ± 0.03	-0.05 ± 0.02	$\mathbf{0.00 \pm 0.02}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
PPO Hash-Count	0.02 ± 0.08	$\mathbf{0.28 \pm 0.07}$	0.27 ± 0.04	0.11 ± 0.07	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
DeA2C Dict-Count	$\mathbf{0.84 \pm 0.06}$	0.62 ± 0.18	0.37 ± 0.21	$\mathbf{0.84 \pm 0.08}$	0.56 ± 0.06	0.00 ± 0.00	0.00 ± 0.00
DePPO Dict-Count	-0.06 ± 0.30	-0.10 ± 0.10	-0.07 ± 0.05	$\mathbf{0.84 \pm 0.07}$	0.56 ± 0.06	0.55 ± 0.09	0.00 ± 0.00
DeDQN Dict-Count	$\mathbf{0.26 \pm 0.25}$	0.20 ± 0.35	0.08 ± 0.38	-0.15 ± 0.03	-0.14 ± 0.06	-0.05 ± 0.02	-0.07 ± 0.01

Table 24: Maximum evaluation returns in Hallway $N_l = N_r = 10$ with prediction-based intrinsic rewards and varying learning rates.

Algorithm \ Learning rate	1e-09	1e-08	2e-08	1e-07	5e-07	1e-06	1e-05	0.0001	0.001
A2C ICM	-0.07 ± 0.09	-0.07 ± 0.09	-0.07 ± 0.09	0.28 ± 0.40	$\mathbf{0.88 \pm 0.74}$	0.85 ± 0.00	0.85 ± 0.00	0.85 ± 0.00	0.28 ± 0.40
A2C RND	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-0.09 ± 0.08	-0.10 ± 0.08	$\mathbf{0.28 \pm 0.39}$
A2C RIDE	-0.11 ± 0.01	-0.11 ± 0.01	-0.11 ± 0.01	0.83 ± 0.02	0.83 ± 0.02	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.53 ± 0.45	0.00 ± 0.00
PPO ICM	0.54 ± 0.38	0.20 ± 0.43	0.47 ± 0.47	0.47 ± 0.47	0.23 ± 0.40	0.20 ± 0.43	0.43 ± 0.41	$\mathbf{0.83 \pm 0.02}$	0.57 ± 0.40
PPO RND	$\mathbf{0.80 \pm 0.00}$	$\mathbf{0.80 \pm 0.00}$	0.54 ± 0.38	0.53 ± 0.38	0.47 ± 0.44	0.50 ± 0.43	0.48 ± 0.48	0.23 ± 0.41	0.53 ± 0.38
PPO RIDE	0.51 ± 0.43	0.50 ± 0.43	$\mathbf{0.80 \pm 0.00}$	0.32 ± 0.44	0.24 ± 0.41	0.55 ± 0.39	0.51 ± 0.43	0.53 ± 0.44	0.28 ± 0.40
DeA2C ICM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.82 ± 0.02	0.57 ± 0.40	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	$\mathbf{0.85 \pm 0.00}$	0.57 ± 0.40
DePPO ICM	-0.07 ± 0.09	-0.07 ± 0.09	-0.07 ± 0.09	0.53 ± 0.45	0.57 ± 0.40	0.57 ± 0.40	$\mathbf{0.85 \pm 0.00}$	0.55 ± 0.39	0.00 ± 0.00
DeDQN ICM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{1.47 \pm 0.47}$	0.42 ± 0.85	0.25 ± 0.43	0.28 ± 0.40	0.00 ± 0.00	0.00 ± 0.00

Table 25: Average evaluation returns in Hallway $N_l = N_r = 10$ with prediction-based intrinsic rewards and varying learning rates.

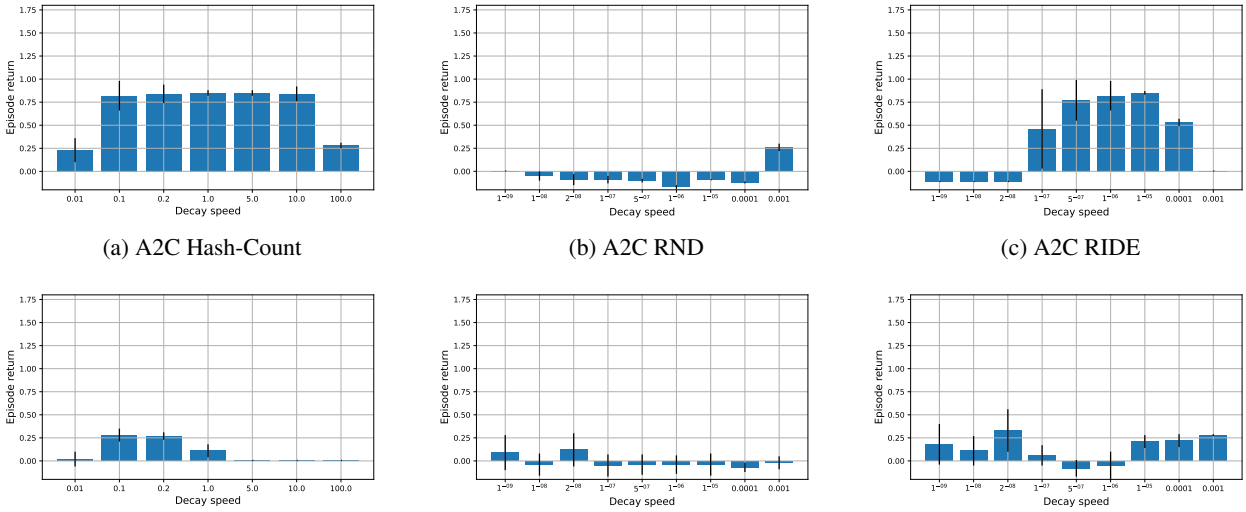
Algorithm \ Learning rate	1e-09	1e-08	2e-08	1e-07	5e-07	1e-06	1e-05	0.0001	0.001
A2C ICM	-0.10 ± 0.01	-0.12 ± 0.04	-0.12 ± 0.03	0.02 ± 0.19	0.47 ± 0.26	0.67 ± 0.22	0.82 ± 0.13	$\mathbf{0.84 \pm 0.09}$	0.28 ± 0.00
A2C RND	0.00 ± 0.00	-0.05 ± 0.05	-0.09 ± 0.06	-0.09 ± 0.04	-0.10 ± 0.02	-0.17 ± 0.02	-0.09 ± 0.00	-0.12 ± 0.01	$\mathbf{0.26 \pm 0.04}$
A2C RIDE	-0.11 ± 0.00	-0.11 ± 0.00	-0.11 ± 0.00	0.46 ± 0.43	0.77 ± 0.22	0.82 ± 0.16	$\mathbf{0.85 \pm 0.02}$	0.53 ± 0.04	0.00 ± 0.00
PPO ICM	0.00 ± 0.19	-0.08 ± 0.07	-0.08 ± 0.11	-0.05 ± 0.16	-0.03 ± 0.13	-0.06 ± 0.07	0.11 ± 0.16	0.30 ± 0.25	$\mathbf{0.48 \pm 0.15}$
PPO RND	0.09 ± 0.19	-0.04 ± 0.12	$\mathbf{0.12 \pm 0.18}$	-0.05 ± 0.12	-0.04 ± 0.11	-0.04 ± 0.10	-0.04 ± 0.12	-0.07 ± 0.05	-0.02 ± 0.07
PPO RIDE	0.18 ± 0.22	0.11 ± 0.16	$\mathbf{0.33 \pm 0.23}$	0.06 ± 0.11	-0.08 ± 0.09	-0.05 ± 0.15	0.21 ± 0.07	0.22 ± 0.07	0.28 ± 0.01
DeA2C ICM	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.39 ± 0.37	0.50 ± 0.15	0.81 ± 0.17	0.73 ± 0.16	$\mathbf{0.84 \pm 0.07}$	0.56 ± 0.03
DePPO ICM	-0.07 ± 0.00	-0.11 ± 0.04	-0.14 ± 0.05	0.14 ± 0.28	0.48 ± 0.21	0.53 ± 0.13	$\mathbf{0.68 \pm 0.18}$	0.47 ± 0.13	0.00 ± 0.00
DeDQN ICM	-0.07 ± 0.02	-0.07 ± 0.02	-0.07 ± 0.02	$\mathbf{0.50 \pm 0.49}$	0.03 ± 0.18	-0.06 ± 0.10	-0.12 ± 0.05	-0.09 ± 0.05	-0.16 ± 0.02



(a) A2C Dict-Count (b) PPO Dict-Count (c) DeA2C Dict-Count (d) DePPO Dict-Count (e) DeDQN Dict-Count

(f) A2C ICM (g) PPO ICM (h) DeA2C ICM (i) DePPO ICM (j) DeDQN ICM

Figure 18: Average evaluation returns for baselines and DeRL with Dict-Count and ICM intrinsic rewards in Hallway $N_l = N_r = 10$ with varying count increments and learning rates.



(a) A2C Hash-Count (b) A2C RND (c) A2C RIDE

(d) PPO Hash-Count (e) PPO RND (f) PPO RIDE

Figure 19: Average evaluation returns for A2C and PPO with Hash-Count, RND and RIDE intrinsic rewards in Hallway $N_l = N_r = 10$ with varying count increments and learning rates.