Internally Rewarded Reinforcement Learning

Mengdi Li[*1] Xufeng Zhao[*1] Jae Hee Lee[1] Cornelius Weber[1] Stefan Wermter[1]

Abstract

We study a class of reinforcement learning problems where the reward signals for policy learning are generated by a discriminator that is dependent on and jointly optimized with the policy. This interdependence between the policy and the discriminator leads to an unstable learning process because reward signals from an immature discriminator are noisy and impede policy learning, and conversely, an under-optimized policy impedes discriminator learning. We call this learning setting *Internally Rewarded Reinforcement Learning* (IRRL) as the reward is not provided directly by the environment but *internally* by the discriminator. In this paper, we formally formulate IRRL and present a class of problems that belong to IRRL. We theoretically derive and empirically analyze the effect of the reward function in IRRL and based on these analyses propose the clipped linear reward function. Experimental results show that the proposed reward function can consistently stabilize the training process by reducing the impact of reward noise, which leads to faster convergence and higher performance compared with baselines in diverse tasks. [2]

1. Introduction

Rewards are essential for animals and artificial agents to learn by exploration in an environment. In the brain, reward signals are emitted by specific neurons as a consequence of the processing of external stimuli (Olds & Milner, 1954; Schultz, 2015). For instance, when a child receives words of praise from the parents as feedback for exhibiting appropriate behavior, the rewards obtained are contingent upon the child's individual understanding of the words. In some cases, the child may misunderstand the praise as criticism, thus wrongly obtaining a negative reward and impeding its behavior learning. An elaborated view of the standard agent-environment interaction formulation (Sutton & Barto, 1998) of reinforcement learning (RL) demonstrates this mechanism (Singh et al., 2004). This framework separates the environment into an *external environment*, which provides external stimuli (e.g., a word of praise from the parents), and an *internal environment*, which is in the same organism with the agent and contains a critic that produces reward signals based on both external stimuli and the internal state (cf. Fig 1 left panel).

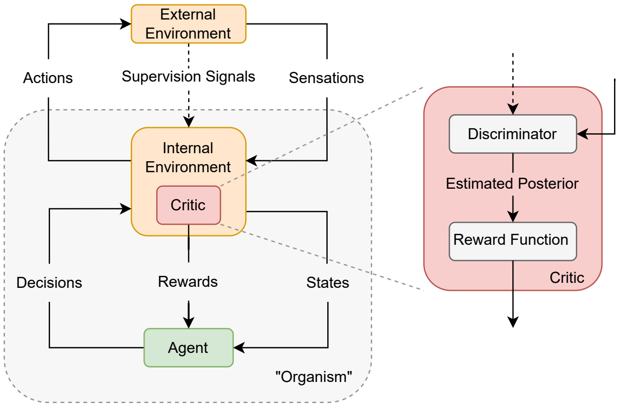


Figure 1: **Left:** The agent-environment interaction loop of IRRL. This diagram is based on the scheme of intrinsically motivated RL (Singh et al., 2004) with an optional path of supervision signals, which reflects an extrinsic reward. **Right:** The internal critic consists of a *discriminator*, which estimates a posterior probability of correct discrimination given sensations and supervision signals from the external environment, and a *reward function*, which produces rewards by processing the posterior probability.

In this work, we focus on situations where the reward is determined by both external stimuli and the state of a sophisticated and evolutionary internal environment that produces either task-relevant rewards (Mnih et al., 2014; Ba et al., 2015; Li et al., 2021; Rangrej et al., 2022) or task-agnostic rewards (Gregor et al., 2017; Strouse et al., 2022), and we use the term *Internally Rewarded Reinforcement Learning* (IRRL) to refer to the learning problem in these situations

[*] Equal contribution [1] Knowledge Technology Group, Department of Informatics, University of Hamburg, Hamburg, Germany. Correspondence to: Mengdi Li <mli@informatik.uni-hamburg.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[2] Project page: https://ir-rl.github.io/

(cf. some IRRL examples in Fig. 3).

In IRRL, the policy of the agent is trained by RL, and the critic of the internal environment is simultaneously trained either in a self-supervised learning (SSL) manner by directly using the sensations from the external environment (Pathak et al., 2017; Gregor et al., 2017; Eysenbach et al., 2019; Strouse et al., 2022), or in a supervised learning (SL) manner by using extra human-annotated task-relevant signals (Mnih et al., 2014; Yu et al., 2017; Tan et al., 2020; Li et al., 2021). The critic provides reward signals for training a policy that, in return, controls the collection of the trajectories for the critic. These scenarios have become prevalent with increased interest in integrating the capability of high-level prediction and low-level control of behaviors into a single model in the realms of attention mechanisms (Mnih et al., 2014; Ba et al., 2015; Yu et al., 2017; Li et al., 2017; Rangrej et al., 2022), embodied agents (Gordon et al., 2018; Yang et al., 2019), robotics (Lakomkin et al., 2018; Li et al., 2021), and unsupervised RL (Gregor et al., 2017; Eysenbach et al., 2019; Strouse et al., 2022).

The role of the critic depends on the target task. In the task of digit recognition with hard attention (see Fig. 3a), for example, the critic assesses the certainty of performing correct digit classification. In the unsupervised skill discovery task (see Fig. 3b), however, the critic works as an intrinsic motivation system to evaluate the novelty of generated skills. The critic consists of a discriminator and a reward function, as shown in the right panel of Fig. 1. The discriminator estimates the posterior probability of the target label provided by supervision signals or sensations. By processing the posterior, the reward function produces rewards for the behavior learning of the agent.

Simultaneous optimization between the policy and the discriminator in IRRL is however non-trivial because of the unstable training loop where neither of them can learn efficiently (see Fig. 2). In this work, we seek to solve this issue by reducing the impact of reward noise, which is challenging due to the unavailability of an oracle discriminator whose posterior probability can reflect the information sufficiency for discrimination. We theoretically formulate IRRL to explicitly analyze the noisy reward issue and characterize the distribution of the noise empirically by approximating the oracle discriminator with the discriminator of a converged model. Based on our formulation and empirical results, we demonstrate the effect of the reward function in reducing the bias of the estimated reward and the variance of the reward noise, and propose a simple yet effective reward function that stabilizes the training process.

We present extensive experimental results on IRRL tasks with task-relevant rewards (i.e., visual hard attention, and robotic active vision), or tasks with task-agnostic rewards (i.e., unsupervised skill discovery). The results suggest that our proposed reward function consistently improves the stability and the speed of training, and achieves better performance than the baselines on all the tasks. In particular, on the skill discovery task, our approach with the simple reward function achieves the same performance as the state-of-the-art sophisticated ensemble-based Bayesian method by Strouse et al. (2022) but without using ensembles. We further demonstrate that the superiority of the proposed reward function is due to its effectiveness in noise reduction, which is in line with our theoretical analysis. The contributions of this paper are summarized as follows:

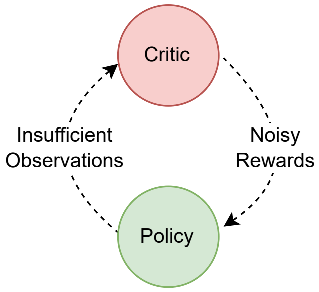


Figure 2: Simultaneous optimization between the policy of the agent and the critic of the internal environment is challenging because an under-optimized critic yields noisy rewards, and in turn, an immature policy yields insufficient observations, which leads to an unstable training loop.

1. We formulate a class of RL problems as IRRL, and formulate the inherent issues of *noisy rewards* that leads to an unstable training loop in IRRL.

2. We empirically characterize the noise in the discriminator and derive the effect of the reward function in reducing the bias of the estimated reward and the variance of the reward noise stemming from an underdeveloped discriminator.

3. We propose a simple yet effective reward function, the *clipped linear reward function*, which consistently stabilizes the training process and achieves faster convergence speed and higher performance on diverse IRRL tasks.

2. Related Work

The RL process is notoriously unstable. Previous work has studied various techniques to stabilize training, such as reducing the bias and variance of gradient estimation for policy gradient methods (Greensmith et al., 2004; Schulman et al., 2015), and value estimation for value-based methods (van Hasselt et al., 2016). As another factor impacting RL training, reward noise that stems from various sources, e.g., sensors on robots, and adversarial attacks, is attracting attention because of the growing interest in applying RL to more realistic and complicated tasks (Huang et al., 2017; Everitt et al., 2017; Wang et al., 2020). In cases where the noise directly resides in the reward, both policy gradient and value-based RL methods suffer. Everitt et al. (2017) and Wang et al. (2020) formulate RL with corrupted rewards and partially address the issue for cases with extra knowledge about the noise. Unlike the noise caused by reward corruption, the noise in IRRL comes from a discriminator and is subject to the learning process, so their approaches are not directly applicable to our scenarios in terms of both formulation and experimental emulation.

The issues of unstable training in IRRL have been mentioned in the literature, but they have not been systematically studied. Some works (Mnih et al., 2014; Ba et al., 2015; Li et al., 2017) ignore the impact of the unstable training loop at the expense of the training speed and the performance of the final model. Other works resort to elaborated training strategies, e.g., staged training (Gordon et al., 2018; Yang et al., 2019; Lysa et al., 2022), curriculum training (Das et al., 2018; Li et al., 2021), imitation learning (Tan et al., 2020; Rangrej et al., 2022), or task-specific reward shaping (Deng et al., 2021). However, extra efforts such as data collection or human ingenuity are needed in these methods.

Strouse et al. (2022) study the pessimistic exploration problem in the context of unsupervised skill discovery (cf. Fig. 3b) where a skill discriminator is used to generate rewards. As the skill discriminator is subject to noise, this issue can be seen as a consequence of the unstable training loop under the framework of IRRL. Similar to our work, they also resort to modifying the reward function. They propose to train an ensemble of discriminators and reward the policy with their disagreement. Experimental results suggest that the proposed disagreement-based reward lets the agent learn more skills through optimistic exploration. However, this method introduces more model parameters and hyper-parameters than baseline methods that are not based on ensembles. In this paper, we consider the issue in a more general context including but not limited to unsupervised skill discovery, and manage the issue in a more simple and efficient way.

3. Internally Rewarded RL

We formulate the policy learning of IRRL as a Markov decision process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p_{\mathrm{E}}, \rho, r, \gamma \rangle$, where, $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $p_{\mathrm{E}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ the state transition probability, $\rho : \mathcal{S} \to \mathbb{R}$ the distribution of the initial state, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward on each transition, and $\gamma \in (0, 1)$ a discount factor.

Different from conventional RL settings, where reward r depends exclusively on the *external* environment, in IRRL reward r is determined by a *critic*, which resides in the *internal* environments and interprets the supervision signals from the external environment to generate internal rewards (cf. Fig. 1). Here, we assume that the external environment, hence the observations an agent is making, is caused by a label y sampled from a prior distribution $p(y)$. The critic depends on a trainable *discriminator* q_ϕ parameterized with ϕ. Given a trajectory $\tau \in (\mathcal{S} \times \mathcal{A})^n$ ($n \in \mathbb{N}$ is the trajectory length) sampled from a policy π_θ parameterized with θ, the discriminator $q_\phi(y \mid \tau)$ computes the probability of the label y being the cause of the trajectory τ.[3]

Many existing works, which have been studied independently before, can be categorized as instances of IRRL. In the following, we present three lines of existing works as concrete examples of IRRL:

1. **Hard attention**. Hard attention mechanism (Mnih et al., 2014; Ba et al., 2015; Li et al., 2017; Rangrej et al., 2022) is essential when all available information is expensive or unrealistic to process, e.g., scene classification for high-resolution satellite images (Wang et al., 2019; Rangrej et al., 2022). Fig. 3a shows the task of hard attention for digit recognition on the Cluttered MNIST dataset (Mnih et al., 2014).

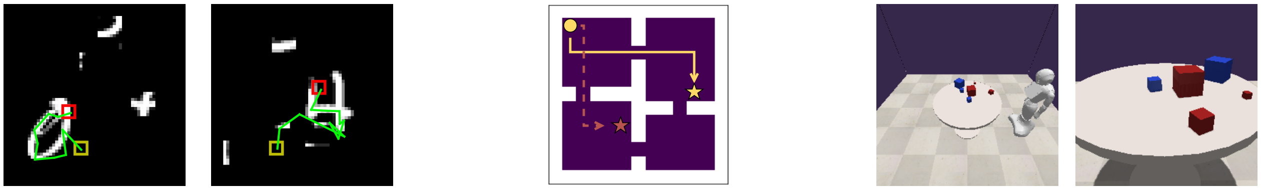
2. **Intrinsically motivated RL**. In this setting, an agent is trained using dense intrinsic rewards to explore the environment based on its curiosity about encountered states (Pathak et al., 2017) or to discover diverse skills based on their novelty (Gregor et al., 2017; Eysenbach et al., 2019; Strouse et al., 2022). Fig. 3b shows the task of unsupervised skill discovery in a four-room environment (Strouse et al., 2022).

3. **Task-oriented active vision**. This is an emerging research topic with the goal of endowing embodied agents with high-level perception and reasoning capabilities. The agent actively changes its egocentric view to collect information for achieving downstream tasks, e.g., question answering (Gordon et al., 2018; Deng et al., 2021; Li et al., 2021), object recognition (Yang et al., 2019), or scene description (Tan et al., 2020). Fig. 3c shows the task of robotic object counting in occlusion scenarios.

3.1. Optimization

In IRRL, the policy and the discriminator are optimized simultaneously with different optimization objectives.

[3] To simplify notations, we use lower-case letters (e.g., y) to both represent random variables and their realizations if the distinction is clear from the context. Similarly, we use $p(y)$ to both represent the distribution of y and the probability of y if the context is clear.



(a) **Hard attention for digit recognition on the Cluttered MNIST dataset** (Mnih et al., 2014). A small glimpse (the squares) controlled by an attention policy sequentially changes its location to collect information for recognizing the digit. During training, the critic is expected to produce rewards that reflect the sufficiency of information collected by the attention policy, and in turn, the policy is expected to attend to informative regions, i.e., pixels of the digit, to collect information for the classifier to learn digit recognition. The starting and stopping glimpses are represented by yellow and red boxes respectively. The green line indicates the positions of intermediate glimpses.

(b) **Unsupervised skill discovery in a four-room environment** (Strouse et al., 2022). An agent spawned at the top-left corner is expected to learn a navigation policy that performs distinguishable skills without using any extrinsic rewards. In this task, a skill is represented by the final state of a trajectory. During training, the agent generates a trajectory conditioned on a randomly sampled skill label, and a discriminator estimates the posterior probability of the trajectory being the target skill, based on which the reward is produced. The policy and the discriminator are optimized simultaneously.

(c) **Robotic object counting in occlusion scenarios**. A humanoid robot is trained to learn a locomotion policy to explore occluded space by rotating around the table and to terminate exploration to achieve efficient counting of specified objects, e.g., *small_blue_cube*. The robot performs the task solely based on its egocentric RGB view. During training, the policy uses the reward that is produced by a critic containing an object counter, which is simultaneously updated with the policy. Similar to the task of hard attention, the reward should be able to evaluate the information sufficiency of observations for correct object counting.

Figure 3: Example tasks of IRRL

3.1.1. Policy Optimization

The optimization objective of policy learning in IRRL can be formulated from two perspectives, which are accuracy maximization and mutual information maximization.

Accuracy maximization. This is an intuitive formulation, where the policy of the agent is optimized to maximize the expectation of an accuracy-based reward

$$r_{\text{acc}} = \mathbb{1}_y \left[\operatorname*{argmax}_{y' \in \mathcal{Y}} q_\phi(y' \mid \tau) \right], \tag{1}$$

where $\mathcal{Y}$ is a set of possible labels and $\mathbb{1}_y[x]$ is an indicator function that returns 1 if x is the target label y, 0 otherwise. This formulation has been widely used in existing works on hard attention (Mnih et al., 2014; Kingma & Ba, 2015; Li et al., 2017), embodied agents (Gordon et al., 2018; Yang et al., 2019), and robotics (Lakomkin et al., 2018; Li et al., 2021). However, an obvious disadvantage of the accuracy-based reward is that it cannot faithfully reflect the discriminator's uncertainty about the observations collected by the reinforcement learner, which makes learning slow and leads to suboptimal performance (cf. Sec. 5). Therefore, it will be analyzed only empirically in this paper.

Mutual information maximization. Mutual information is commonly used to estimate the relationship between pairs of random variables. The objective of mutual information maximization has been utilized in the realm of unsupervised skill discovery (Gregor et al., 2017; Eysenbach et al., 2019; Strouse et al., 2022). We generalize it to the optimization objective of IRRL.

Given a target label y and a trajectory τ sampled from $p(y)$ and π_θ respectively, their mutual dependency can be obtained by the KL-divergence of their joint distribution $p(y, \tau)$ and the product of their marginal distributions $p(y)p(\tau)$:

$$\begin{aligned} I(y;\tau) &:= D_{\mathbb{KL}}(p(y,\tau) \parallel p(y)p(\tau)) \\ &= \mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)} \left[\log p(y \mid \tau) - \log p(y)\right], \end{aligned} \tag{2}$$

which is also known as Shannon's mutual information between y and τ and which reaches its maximum if the full knowledge of y can be deduced from τ. In this equation, $p(y \mid \tau)$ is the oracle posterior probability that reflects the information sufficiency of observations for discrimination. It can be interpreted as being generated by an *oracle discriminator*, a conceptual term utilized for the theoretical formulation. If $p(y \mid \tau)$ is known, then by defining

$$r^*_{\log} := \log p(y \mid \tau) - \log p(y) \tag{3}$$

as the reward for an RL algorithm involving π_θ, one can maximize $I(y;\tau)$, i.e., π_θ generates trajectories for an optimal discrimination of the target label y.

Because the oracle discriminator $p(y \mid \tau)$ is not available in practice, we can replace $p(y \mid \tau)$ with a neural network $q_\phi(y \mid \tau)$ with trainable parameters ϕ and define the reward as

$$r_{\log} = \log q_\phi(y \mid \tau) - \log p(y), \tag{4}$$

and maximize the Barber-Agakov variational lower bound of $I(y;\tau)$ (Barber & Agakov, 2003):

$$I_{\mathrm{BA}}(y;\tau) := \mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)}[\log q_\phi(y \mid \tau) - \log p(y)]. \quad (5)$$

3.1.2. Discriminator Optimization

Concurrent with policy learning, the discriminator $q_\phi(y \mid \tau)$ is trained to better approximate $p(y \mid \tau)$. To this end, instead of the cross-entropy loss

$$-\mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)}\left[p(y \mid \tau)\log q_\phi(y \mid \tau)\right], \quad (6)$$

which involves the oracle discriminator $p(y \mid \tau)$, a proxy cross-entropy loss

$$-\mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)} \log q_\phi(y \mid \tau) \quad (7)$$

is used in practice, which is equivalent to assuming $p(y \,|\, \tau) = 1$, i.e., assuming that τ contains sufficient information for deducing y with the oracle discriminator.

3.2. The Issue of Reward Noise

As the trainable discriminator $q_\phi(y \mid \tau)$ only approximates the oracle discriminator $p(y \mid \tau)$, it inevitably introduces noise $\varepsilon_{\log}$ in the reward $r_{\log}$ in Eq. (4), which is given by

$$\varepsilon_{\log} = r_{\log} - r^\star_{\log} = \log q_\phi(y \mid \tau) - \log p(y \mid \tau). \quad (8)$$

To demonstrate the negative impact of reward noise on the learning process (cf. Fig. 2), we conduct *reward hacking* experiments, where we replace the trainable discriminator $q_\phi(y \mid \tau)$ with a pretrained one $q_{\tilde\phi}(y \mid \tau)$ that is obtained from a converged model to mimic the oracle discriminator $p(y \mid \tau)$. The setup of the reward hacking experiment is illustrated in Fig. 4. We choose the digit recognition task as the target task (cf. Fig. 3a) and use the recurrent attention model (RAM) (Mnih et al., 2014) (detailed information about this task and the model is given in Sec. 5.1 and Appendix F).

Fig. 5 shows a plot of training curves when using different reward functions with and without reward hacking. As shown by the gap between the training curves when using an identical reward function with and without reward hacking, the noise of an under-optimized discriminator influences the training process negatively. In this paper, we aim to narrow the gap by devising an effective reward. As we will see in the next section, a well-designed reward is key to stabilizing the learning process.

4. Reward Noise Moderation

In this section, we first analyze the reduction of the bias of the estimated reward and the variance of the reward noise and then propose a reward that alleviates the negative effect of reward noise and stabilizes the training process.

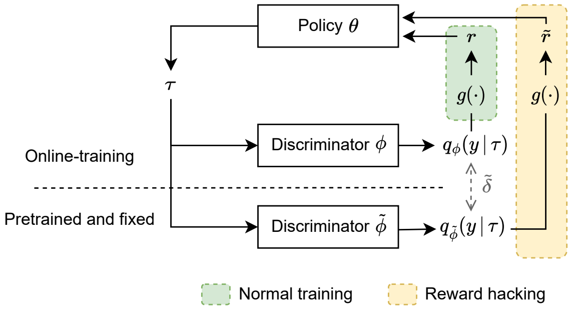


Figure 4: Illustration of the experimental setup of reward hacking. In normal training, the reward is produced based on the posterior probability estimated by an online-training discriminator ϕ. In training with reward hacking, the reward stems from a pretrained and fixed discriminator $\tilde\phi$. $\tilde\delta$ indicates the difference between a pair of posterior probabilities estimated by ϕ and $\tilde\phi$.

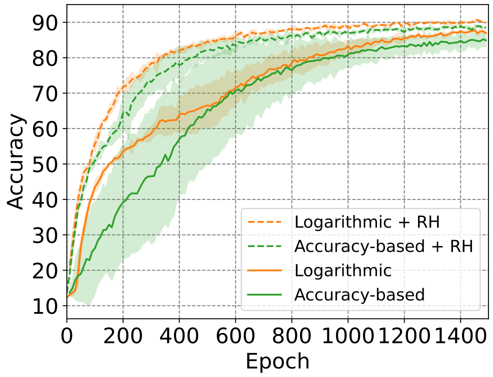


Figure 5: RAM trained using the accuracy-based and the logarithmic reward with and without reward hacking (RH). A model without reward hacking is subject to more noisy rewards and suffers from an unstable learning process, resulting in slower convergence and lower accuracy.

4.1. Generalized Reward

Since the noisy reward in Eq. (4) is a transformation of the posterior probability $q_\phi(y \mid \tau)$, it is reasonable to study the effect of a series of transformations of $q_\phi(y \mid \tau)$ as long as they agree on the same optimal objective. Based on the logarithmic transformation in Eq. (4), the *generalized reward* is defined as

$$r_g = g\left[q_\phi(y \mid \tau)\right] - g\left[p(y)\right] \quad (9)$$

and the *generalized oracle reward* as

$$r^*_g = g\left[p(y \mid \tau)\right] - g\left[p(y)\right], \quad (10)$$

where g is an increasing function (e.g., log), such that maximizing $g(\cdot)$ leads to the maximization of the mutual information $I(y;\tau)$. When selecting the appropriate function, it is important to consider both its ability to transmit information and its ability to moderate noise. The former ensures that the maximization of mutual information can be achieved efficiently, while the latter helps to reduce the impact of reward noise. [4]

4.2. Generalized Reward Noise

To analyze the noise in the generalized reward r_g we apply the second-order Taylor approximation to the *generalized reward noise*

$$\varepsilon_g := r_g - r_g^* = g\left[q_\phi(y \mid \tau)\right] - g\left[p(y \mid \tau)\right] \tag{11}$$

at point $p(y \mid \tau)$. By defining

$$\delta := q_\phi(y \mid \tau) - p(y \mid \tau) \tag{12}$$

as the *discriminator noise*, we have as the expectation of the reward noise (equivalently, the bias of the reward estimator)

$$\mathbb{E}[\varepsilon_g] \approx g'(p(y \mid \tau))\mathbb{E}[\delta] + \frac{1}{2!}g''(p(y \mid \tau))\mathbb{E}\left[\delta^2\right], \tag{13}$$

and as the variance of the reward noise

$$\begin{aligned}\mathbb{V}[\varepsilon_g] \approx (g'(p(y \mid \tau)))^2\mathbb{V}[\delta] + (\frac{1}{2!}g''(p(y \mid \tau)))^2\mathbb{V}\left[\delta^2\right] \\ + g'(p(y \mid \tau))g''(p(y \mid \tau))\mathrm{Cov}\left[\delta, \delta^2\right].\end{aligned} \tag{14}$$

Our goal is to mitigate the impact of the reward noise by minimizing the expectation and the variance of the noise. This is expected to be achieved especially at the early learning stage when the issue of the unstable training loop is severe because both the discriminator and the policy are immature: the trajectory collected by the policy contains little information for discrimination, and the estimated posterior of the discriminator cannot reflect the sufficiency of information collected by the policy. To this end, in the following subsections, we theoretically and empirically analyze Eq. (13) and Eq. (14) and investigate reward functions.

4.3. Characterization of the Discriminator Noise

We make hypotheses regarding the distribution characteristics of δ, which is necessary to analyze the expectation and variance of the reward noise ε_g according to Eq. (13) and Eq. (14). We hypothesize that the expectation of the discriminator noise δ is zero, i.e., $\mathbb{E}[\delta] = 0$, and the distribution of δ is symmetric.

We conduct an empirical study of the distribution of the discriminator noise following the setup of the reward hacking experiment (cf. Fig. 4). Instead of using a pretrained discriminator to interfere in the training process, we visualize the approximated discriminator noise $\tilde{\delta}$ during normal training. $\tilde{\delta}$ is the difference between the posterior probabilities estimated by the online-training discriminator and the pretrained discriminator, i.e., $\tilde{\delta} = q_\phi(y \mid \tau) - q_{\tilde{\phi}}(y \mid \tau) \approx \delta$.

Fig. 6 demonstrates violin plots of the discriminator noise at four training epochs (the model converges at about 1200 epochs). Each violin plot is drawn from 1000 random samples from the testing dataset. We can observe that the mean of the noise is close to zero at different training stages, i.e., $\mathbb{E}[\delta] \approx 0$, and the plots are almost symmetrical with respect to the average noise except at the very beginning when the model weights are being updated after random initialization.

We assume that the noise characteristics are generalized to other problems of IRRL because of the shared high-level abstraction among them (cf. Fig. 1). Characteristics of the discriminator noise when using other reward functions are given in Appendix B.

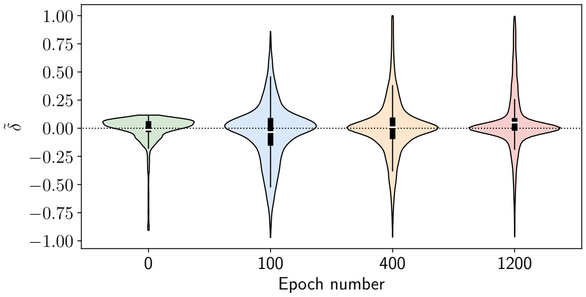


Figure 6: Violin plots of the approximated discriminator noise $\tilde{\delta}$ in the training process of RAM, of which the policy is trained using the logarithmic reward function (cf. Eq. (4)). The small white bar indicates the mean of the noise. The thick vertical line represents the interquartile range and the thin vertical line represents the area between the upper and lower adjacent values.

4.4. Linear Reward

Considering the impact of $g(\cdot)$ on the Taylor approximation to $\mathbb{E}[\varepsilon_g]$ and $\mathbb{V}[\varepsilon_g]$ in Eq. (13) and Eq. (14), we propose a linear reward

$$r_{\text{lin}} = q_\phi(y \mid \tau) - p(y), \tag{15}$$

instead of the commonly applied logarithmic reward $r_{\log}$, to stabilize IRRL. The corresponding expectation and variance of the noise are $\mathbb{E}[\varepsilon_{\text{lin}}] = \mathbb{E}[\delta] = 0$ and $\mathbb{V}[\varepsilon_{\text{lin}}] = \mathbb{V}[\delta]$, respectively. The linear reward enjoys lower reward bias than the logarithmic reward, since

$$|\mathbb{E}[\varepsilon_{\log}]| \approx \frac{1}{2!\,p^2(y \mid \tau)}\mathbb{E}[\delta^2] > 0 = |\mathbb{E}[\varepsilon_{\text{lin}}]|. \tag{16}$$

Furthermore, the variance of r_{lin} is low and stable compared with the variance of logarithmic reward $r_{\log}$, which

[4] The transformation $g(\cdot)$ can also be motivated by the f-mutual information objectives (see Appendix A.2).

suffers from high variance $\mathbb{V}[\varepsilon_{\text{log}}] \approx p^{-2}(y \mid \tau)\mathbb{V}[\delta] + (\frac{1}{2!\,p^2(y|\tau)})^2\mathbb{V}[\delta^2]$ since $p(y \mid \tau) < 1$ in most cases and is dependent on the training policy. (A detailed derivation is given in Appendix A.1. The evaluation of various g functions is given in Appendix D.)

4.5. Clipped Linear Reward

The issue of reward noise is not fully tackled by using the linear reward. Given a target label y, it is intuitive to assume that the posterior probability $p(y \mid \tau)$ of an oracle discriminator should be, in most cases, equal or larger than the prior $p(y)$, as y is a cause of the trajectory τ. However, a discriminator q_ϕ may return a posterior probability $q_\phi(y \mid \tau)$ lower than $p(y)$, especially at the early training stage when both the policy and the discriminator are under-optimized.

Since we expect $q_\phi(y \mid \tau)$ to be close to $p(y \mid \tau)$, we replace the term $q_\phi(y \mid \tau)$ of r_{lin} in Eq. (15) with $\max(q_\phi(y \mid \tau), p(y))$ to integrate the prior knowledge and define

$$\begin{aligned}\overline{r_{\text{lin}}} &:= \max(q_\phi(y \mid \tau), p(y)) - p(y) \\ &= \max(q_\phi(y \mid \tau) - p(y), 0),\end{aligned} \tag{17}$$

which we call the *clipped linear reward.*

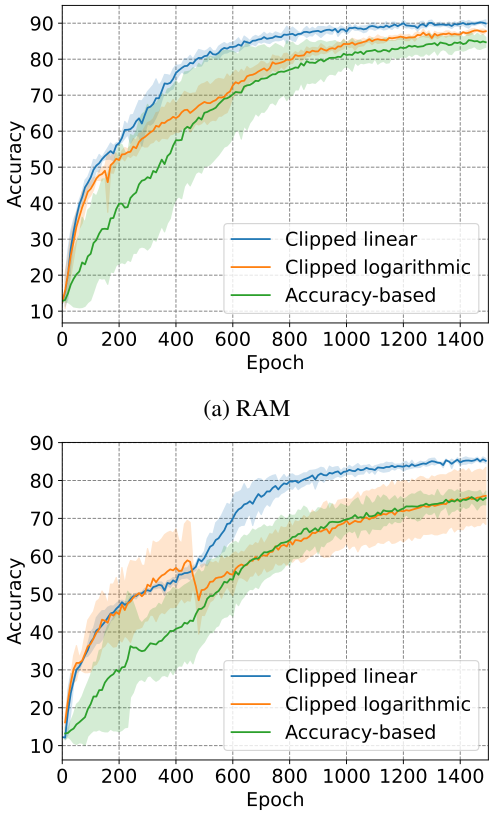
Similar clipping techniques are empirically found to be beneficial when applied to the logarithmic reward (Strouse et al., 2022). In this paper, we go further with an analysis of reward functions from the perspective of noise moderation and achieve better performance with the proposed reward. The proposed clipped linear reward has a similar shape to the rectified linear unit (ReLU) activation function (Nair & Hinton, 2010) which preserves information about relative intensities in multiple layers of deep neural networks. Likewise, the clipped linear reward function can robustly preserve information that travels from an internal discriminator to the policy network.

5. Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed method on the three aforementioned tasks in Sec. 3. We first introduce experimental setups and baselines. Then, we compare the proposed clipped linear function with multiple baselines including state-of-the-art methods. Finally, we conduct reward hacking experiments on the clipped linear reward function to visualize its capability in reducing the impact of reward noise.

5.1. Experimental Setup

Hard attention for digit recognition. We adopt the dataset configuration of Mnih et al. (2014), and use two basic models for this task: the recurrent attention model (RAM) (Mnih et al., 2014) and the dynamic-time recurrent attention model (DT-RAM) (Li et al., 2017). RAM performs a fixed number of movement steps before performing the final digit recognition, while the policy of DT-RAM learns to terminate the exploration before reaching a maximum number of movement steps. The performance of the agent is evaluated using the accuracy of the digit recognition.



(a) RAM

(b) DT-RAM

Figure 7: Comparison between the clipped linear reward function (🟦) with baselines, including the clipped logarithmic (🟧) and the accuracy-based (🟩) reward function, on the task of hard attention for digit recognition using RAM and DT-RAM. All the experiments in this paper ran over three random seeds. Lines and shaded areas show the mean and standard deviation over multiple runs.

Unsupervised skill discovery. We use the same experimental setup and basic model on the four-room environment as in the work of the discriminator disagreement intrinsic reward (DISDAIN) (Strouse et al., 2022). The performance of the agent is evaluated using the number of learned skills.

Robotic object counting. The setup is based on the task of object existence prediction (Li et al., 2021). We use their model and train it using PPO (Schulman et al., 2017) instead of REINFORCE for higher efficiency. The performance of the agent is evaluated using the accuracy of object counting.

Details of the environments and model implementations can be found in Appendix E and F.

5.2. Baselines

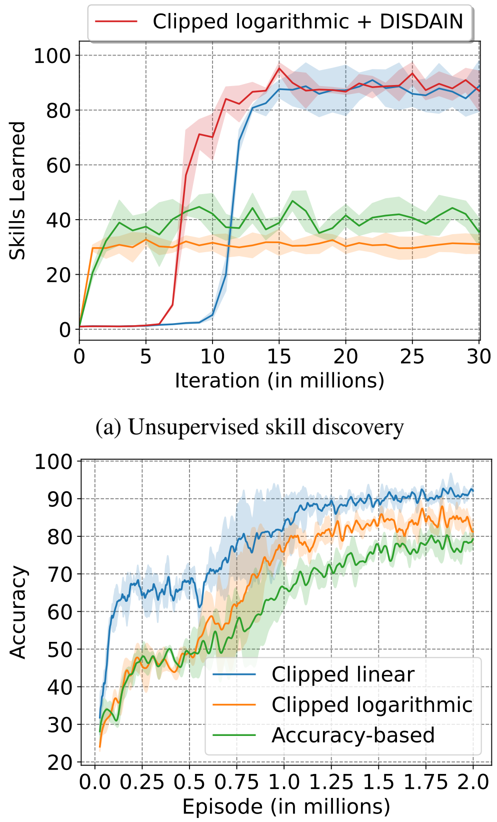
We compare the proposed clipped linear reward function with alternative reward functions. The first is the *accuracy-based reward function* r_{acc} in Eq. (1). The second is the *logarithmic reward function* based on Shannon's mutual information. Instead of using the original logarithmic reward function (Eq. (4)), we use a clipped variant, i.e., $\overline{r_{\log}} = \max(\log q_\phi(y \mid \tau) - \log p(y), 0)$, for fair comparison with our clipped linear reward function. We found that reward clipping generally results in similar or better performance in our experiments, which is consistent with the empirical finding by Strouse et al. (2022). The empirical study of reward clipping is provided in Appendix C. On the skill discovery task, we additionally compare our reward function with the state-of-the-art *DISDAIN reward function* (Strouse et al., 2022) (see Appendix G for details), which was designed specifically to mitigate the pessimistic exploration issue in this task.

5.3. Results

Fig. 7 shows that both RAM and DT-RAM trained using the clipped linear reward function achieve the highest accuracy and fastest training speed. Furthermore, the small blue shaded areas indicate that multiple runs using the clipped linear reward function are consistent with each other, which suggests high stability of the training process.

Fig. 8a demonstrates that the clipped linear reward function outperforms both the clipped logarithmic reward function and the accuracy-based reward function by a large margin and achieves almost the same performance as DISDAIN. We note that the DISDAIN method depends on an ensemble of discriminators and needs more hyper-parameters to tune, e.g., the weight of the DISDAIN reward and the number of ensemble members, while our method is much simpler. Fig. 8b shows that the clipped linear reward function also benefits the challenging robotic object counting task by making the model converge faster and achieve the highest final accuracy.

We can see that the clipped linear reward function generally outperforms the logarithmic and the accuracy-based reward function. The improvement is significant on the skill discovery task, which makes sense according to our theoretical analysis in Sec. 4.4. Since the number of possible discrimination classes in the skill discovery task (128 classes) is much larger than that of other tasks (10 classes in the digit recognition task, and 7 classes in the robotic object counting task), $p(y \mid \tau)$ tends to be closer to zero when trajectory τ contains a small amount of information for discrimination in the skill discovery task. Thus the expectation and variance of the noise of the logarithmic reward function are larger, resulting in a severer unstable training issue, while our clipped linear reward function resulting in low expectation and variance of the reward noise still performs well.



(a) Unsupervised skill discovery

(b) Robotic object counting

Figure 8: Comparison with baselines on the tasks of unsupervised skill discovery and robotic object counting. On the unsupervised skill discovery task, the auxiliary DISDAIN reward (🟥) is compared. Legends are shared between the two sub-figures.

Interesting case studies for the digit recognition and object counting tasks are given in Appendix H.1. An intuitive comparison of state occupancy in the unsupervised skill discovery task is given in Appendix H.2.

5.4. Effect of Noise Moderation

Following the experimental setup in Sec. 3.2, we conduct reward hacking experiments using the clipped linear reward to visualize its capability in narrowing the gap between training processes with and without reward hacking (cf. Fig. 4). Fig. 9 shows the training curves. In order to facilitate a

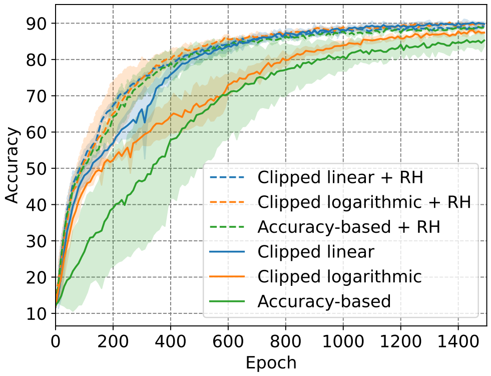


Figure 9: RAM trained using the three kinds of reward functions with and without reward hacking (RH). The clipped linear reward function achieves a much smaller gap between the training processes with and without reward hacking.

comprehensive comparison, we incorporate training curves when using the accuracy-based and the logarithmic reward (cf. Fig. 5) into the figure. We can see from Fig. 9 that when using reward hacking, all three rewards perform similarly (see dashed lines). This suggests that the linear function performs as well as the logarithmic function in terms of information transmission. However, when not using reward hacking, the training curve of the clipped linear reward is much closer to the training curve of using reward hacking, compared to the other two rewards. This suggests that the advantage of the clipped linear reward function is due to the reduction of the impact of reward noise.

6. Discussion

6.1. Interpretation from the Information-theoretic Perspective

The linear reward function has specific meanings from an information-theoretic perspective. It can be derived from the optimization objective of maximizing the χ^2-divergence, one of the f-mutual information measures (Csiszár, 1972; Esposito et al., 2020), instead of the commonly used KL-divergence corresponding to Shannon's mutual information (Shannon, 1948) (cf. Eq. (2)). The derivation is provided in Appendix A.2. In recent years, f-mutual information has been studied in many deep learning applications, such as generative models (Nowozin et al., 2016; Gimenez & Zou, 2022), representation learning (Lotfi-Rezaabad & Vishwanath, 2020; Abstreiter et al., 2021), image classification (Wei & Liu, 2021), imitation learning (Zhang et al., 2020), etc. Wei & Liu (2021) suggested that a properly defined f-divergence measure is robust with label noise in a classification task, which is related to our finding that the χ^2-mutual information is a more robust information measure against the inherent noise in the policy learning of IRRL compared to Shannon's mutual information. This leads to interesting future work on investigating principles for selecting the optimal f-mutual information measure, and the possibility of using other f-mutual information measures for achieving more stable IRRL.

6.2. Limitations and Future Work

This work is an early step towards stabilizing IRRL. Some identified limitations potentially lead to interesting future work. First, we only consider classification-based critics but not regression-based critics. A unified guideline for designing reward functions in both cases is appealing and significant. Second, we stabilize the training process of IRRL from the perspective of reducing the impact of reward noise without explicitly considering reducing the impact of insufficient observations (see Fig. 1). To alleviate the impact of insufficient observations, for example, one can assess the sufficiency of observations according to the consistency of discriminators of an ensemble and remove or replace the outliers. An integrated method considering both issues should lead to a more optimal solution.

7. Conclusion

In this work, we formulate a class of RL problems with internally rewarded RL where a policy and a discriminator functionally interact with each other and are simultaneously optimized. The inherent issues of noisy rewards and insufficient observations in the training process lead to an unstable training loop where neither the policy nor the discriminator can learn effectively. Based on theoretical analysis and empirical studies, we propose the clipped linear reward function to reduce the impact of reward noise. Extensive experimental results suggest that the proposed method can consistently stabilize the training process and achieve faster convergence and higher performance compared with baselines in diverse tasks. Additionally, we give an interpretation of the use of the linear reward function from the information-theoretic perspective, which suggests interesting future work. As interest grows in integrating the capability of high-level prediction and low-level control of behaviors into a single model, for instance in embodied AI, robotics, and unsupervised RL, stable and efficient training of IRRL will be particularly relevant. We hope this work paves the way to achieving this goal.

Acknowledgements

We gratefully acknowledge support from the China Scholarship Council (CSC) and the German Research Foundation (DFG) under project Crossmodal Learning (TRR 169).

References

Abstreiter, K., Mittal, S., Bauer, S., Schölkopf, B., and Mehrjou, A. Diffusion-based representation learning. *arXiv preprint arXiv:2105.14257*, 2021.

Ba, J., Mnih, V., and Kavukcuoglu, K. Multiple object recognition with visual attention. In *International Conference on Learning Representations (ICLR)*, 2015.

Barber, D. and Agakov, F. Information maximization in noisy channels: A variational approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 201–208, 2003.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 530–539, 2018.

Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T. H., and Bengio, Y. BabyAI: First steps towards grounded language learning with a human in the loop. In *International Conference on Learning Representations (ICLR)*, 2019.

Csiszár, I. A class of measures of informativity of observation channels. *Periodica Mathematica Hungarica*, 2: 191–213, 1972.

Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., and Batra, D. Embodied question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–10, 2018.

Deng, Y., Guo, D., Guo, X., Zhang, N., Liu, H., and Sun, F. MQA: Answering the question via robotic manipulation. In *Robotics: Science and Systems (RSS)*, 2021.

Elsayed, G. F., Kornblith, S., and Le, Q. V. Saccader: Improving accuracy of hard attention models for vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 700–712, 2019.

Esposito, A. R., Gastpar, M., and Issa, I. Robust generalization via f-mutual information. In *IEEE International Symposium on Information Theory (ISIT)*, pp. 2723–2728, 2020.

Everitt, T., Krakovna, V., Orseau, L., and Legg, S. Reinforcement learning with a corrupted reward channel. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4705–4713, 2017.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations (ICLR)*, 2019.

Gimenez, J. R. and Zou, J. Y. A unified f-divergence framework generalizing VAE and GAN. *arXiv preprint arXiv:2205.05214*, 2022.

Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., and Farhadi, A. IQA: visual question answering in interactive environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4089–4098, 2018.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 5:1471–1530, 2004.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. In *International Conference on Learning Representations (ICLR Workshops)*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Huang, S. H., Papernot, N., Goodfellow, I. J., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. In *International Conference on Learning Representations (ICLR)*, 2017.

Kapturowski, S., Ostrovski, G., Quan, J., Munos, R., and Dabney, W. Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Kinney, J. B. and Gurinder S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences (PNAS)*, 111(9):3354–3359, 2014.

Lakomkin, E., Zamani, M., Weber, C., Magg, S., and Wermter, S. EmoRL: Continuous acoustic emotion classification using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–6, 2018.

Li, M., Weber, C., Kerzel, M., Lee, J. H., Zeng, Z., Liu, Z., and Wermter, S. Robotic occlusion reasoning for efficient object existence prediction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2686–2692, 2021.

Li, Z., Yang, Y., Liu, X., Zhou, F., Wen, S., and Xu, W. Dynamic computational time for visual attention. In *IEEE International Conference on Computer Vision (ICCV Workshops)*, pp. 1199–1209, 2017.

Lotfi-Rezaabad, A. and Vishwanath, S. Learning representations by maximizing mutual information in variational autoencoders. In *IEEE International Symposium on Information Theory (ISIT)*, pp. 2729–2734, 2020.

Lysa, Y., Weber, C., Becker, D., and Wermter, S. Word-by-word generation of visual dialog using reinforcement learning. In *International Conference on Artificial Neural Networks (ICANN)*, pp. 123–135, 2022.

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2204–2212, 2014.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 271–279, 2016.

Olds, J. and Milner, P. Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain. *Journal of Comparative and Physiological Psychology*, 47(6):419, 1954.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 70, pp. 2778–2787, 2017.

Peng, J. and Williams, R. J. Incremental multi-step Q-learning. *Machine Learning*, 22(1-3):283–290, 1996.

Rangrej, S. B., Srinidhi, C. L., and Clark, J. J. Consistency driven sequential transformers attention model for partially observable scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2508–2517, 2022.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2015.

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

Schultz, W. Neuronal reward and decision signals: From theories to data. *Physiological Reviews*, 95(3):853–951, 2015.

Shannon, C. E. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Singh, S., Barto, A. G., and Chentanez, N. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1281–1288, 2004.

Sønderby, S. K., Sønderby, C. K., Maaløe, L., and Winther, O. Recurrent spatial transformer networks. *arXiv preprint arXiv:1509.05329*, 2015.

Strouse, D., Baumli, K., Warde-Farley, D., Mnih, V., and Hansen, S. Learning more skills through optimistic exploration. In *International Conference on Learning Representations (ICLR)*, 2022.

Sutton, R. S. and Barto, A. G. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

Tan, S., Liu, H., Guo, D., Zhang, X., and Sun, F. Towards embodied scene description. In *Robotics: Science and Systems (RSS)*, 2020.

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

Wang, J., Liu, Y., and Li, B. Reinforcement learning with perturbed rewards. In *AAAI Conference on Artificial Intelligence*, pp. 6202–6209, 2020.

Wang, Q., Liu, S., Chanussot, J., and Li, X. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1155–1167, 2019.

Wei, J. and Liu, Y. When optimizing f-divergence is robust with label noise. In *Conference on Learning Representations (ICLR)*, 2021.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Yang, J., Ren, Z., Xu, M., Chen, X., Crandall, D. J., Parikh, D., and Batra, D. Embodied amodal recognition: Learning to move to perceive objects. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2040–2050, 2019.

Yu, A. W., Lee, H., and Le, Q. V. Learning to skim text. In Barzilay, R. and Kan, M. (eds.), *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1880–1890, 2017.

Zhang, X., Li, Y., Zhang, Z., and Zhang, Z. f-GAIL: Learning f-divergence for generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 12805–12815, 2020.

A. Proofs

A.1. Noise of the Logarithmic Reward

Based on the formulation of Eq. (13) and Eq. (14), the expectation and variance of the reward noise when using the logarithmic reward ($\varepsilon_{\log}$) can be derived as follows:

$$\begin{aligned}
\mathbb{E}[\varepsilon_{\log}] &= g'(p(y \mid \tau))\mathbb{E}[\delta] + \frac{1}{2!}g''(p(y \mid \tau))\mathbb{E}[\delta^2] + \mathbb{E}[o(\delta^2)] \\
&= \frac{1}{p(y \mid \tau)}\mathbb{E}[\delta] - \frac{1}{2!\, p^2(y \mid \tau)}\mathbb{E}[\delta^2] + \mathbb{E}[o(\delta^2)] \\
&\approx -\frac{1}{2!\, p^2(y \mid \tau)}\mathbb{E}[\delta^2],
\end{aligned}$$

$$\begin{aligned}
\mathbb{V}[\varepsilon_{\log}] &= (g'(p(y \mid \tau)))^2\mathbb{V}[\delta] + (\frac{1}{2!}g''(p(y \mid \tau)))^2\mathbb{V}[\delta^2] + g'(p(y \mid \tau))g''(p(y \mid \tau))\mathrm{Cov}[\delta, \delta^2] \\
&\quad + \mathbb{V}[o(\delta^2)] + 2g'(p(y \mid \tau))\mathrm{Cov}[\delta, o(\delta^2)] + g''(p(y \mid \tau))\mathrm{Cov}[\delta^2, o(\delta^2)] \\
&\approx (g'(p(y \mid \tau)))^2\mathbb{V}[\delta] + (\frac{1}{2!}g''(p(y \mid \tau)))^2\mathbb{V}[\delta^2] + g'(p(y \mid \tau))g''(p(y \mid \tau))\mathrm{Cov}[\delta, \delta^2] \\
&\approx (g'(p(y \mid \tau)))^2\mathbb{V}[\delta] + (\frac{1}{2!}g''(p(y \mid \tau)))^2\mathbb{V}[\delta^2] \\
&= \frac{1}{p^2(y \mid \tau)}\mathbb{V}[\delta] + (\frac{1}{2!\, p^2(y \mid \tau)})^2\mathbb{V}[\delta^2].
\end{aligned}$$

The variance is approximated using the fact that $\mathrm{Cov}[\delta, \delta^2] = \mathbb{E}[\delta^3] - \mathbb{E}[\delta]\mathbb{E}][\delta^2] = (\mu^3 + 3\mu\sigma^2 + \gamma\sigma^3) - \mu(\mu^2 + \sigma^2) = 2\mu\sigma^2 + \gamma\sigma^3 \approx 0$, where $\sigma^2 = \mathbb{E}[(\delta - \mu)^2]$ is the variance, and $\mu = \mathbb{E}[\delta]$ and $\gamma = \mathbb{E}[(\frac{\delta-\mu}{\sigma})^3]$ are the mean and skewness, which are both about zero due to the symmetry of the distribution of δ (Sec. 4.3).

A.2. Derivation of the Linear Reward Function by Maximizing the χ^2-divergence

We use the optimization objective of maximizing the f-mutual information between the observation trajectory τ and the target class y in place of the objective of Eq. (2) and obtain

$$\begin{aligned}
I_f(y; \tau) &:= D_f(p(y, \tau) \,||\, p(y)p(\tau)) \\
&= \mathbb{E}_{\tau \sim \pi_\theta, y \sim p(y)} F\left(\frac{p(y \mid \tau)}{p(y)}\right),
\end{aligned} \tag{18}$$

where $D_f(P \,||\, Q) := \mathbb{E}_{q(x)} f\left(\frac{p(x)}{q(x)}\right) = \mathbb{E}_{p(x)} F\left(\frac{p(x)}{q(x)}\right)$ is the f-divergence of two probability distributions P and Q on X, with $f : \mathbb{R}^+ \to \mathbb{R}$ being a generic convex function satisfying $f(1) = 0$, $F(x) := f(x)/x$ for simplicity of expectation over P instead of Q for later use, and $p(x)$ and $q(x)$ are probability density functions of P and Q respectively. By choosing $f(x) = x \log x$, f-divergence becomes the well-known Kullback–Leibler divergence and, correspondingly, the f-mutual information is then Shannon's mutual information (Shannon, 1948; Kinney & Gurinder S. Atwal, 2014; Belghazi et al., 2018). Other typically used f-divergences and their expected mutual information over $p(x, y)$ are listed in Table 1. When using the χ^2-divergence, i.e., $f(x) = (x - 1)^2$, f-mutual information becomes

$$I_f(y; \tau) = \mathbb{E}_{\tau \sim \pi_\theta, y \sim p(y)} \left[\frac{p(y \mid \tau)}{p(y)} - 1\right]. \tag{19}$$

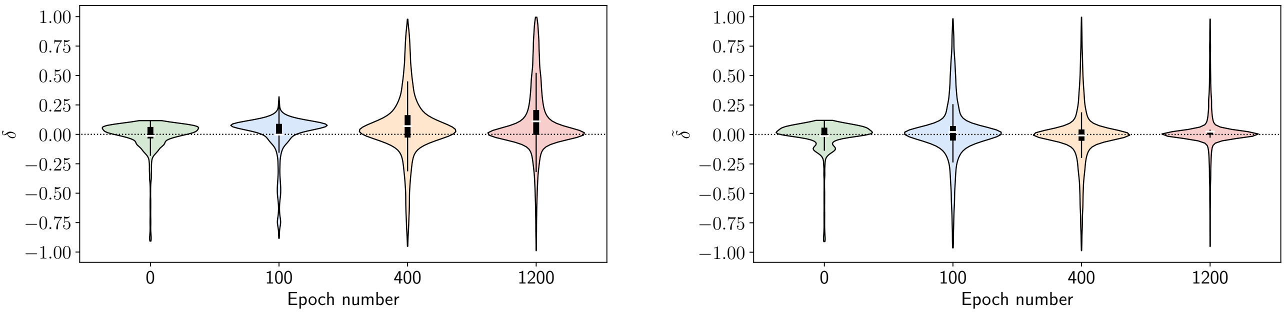
When y is sampled from a uniform distribution, i.e., $p(y)$ is a constant, we have $I_f(y; \tau) = \alpha\, \mathbb{E}_{\tau \sim \pi_\theta, y \sim p(y)} \left[p(y \mid \tau) - p(y)\right]$, where $\alpha = 1/p(y)$. Following the derivation of Eq. (3), this optimization objective induces the linear reward function in Sec. 4.4.

Table 1: f-mutual information and the corresponding convex functions

f-divergence	$f(x)$	$I_f(x;y)$
Kullback–Leibler	$x\log x$	$\mathbb{E}_{p(x,y)}\log\frac{p(y\mid x)}{p(y)}$
χ^2	$(x-1)^2$	$\mathbb{E}_{p(x,y)}\frac{p(y\mid x)}{p(y)}-1$
Total Variance	$\frac{1}{2}\lvert x-1\rvert$	$\mathbb{E}_{p(x,y)}\frac{1}{2}\left\lvert 1-\frac{p(y)}{p(y\mid x)}\right\rvert$
Squared Hellinger	$(1-\sqrt{x})^2$	$\mathbb{E}_{p(x,y)}\left[2-2\sqrt{\frac{p(y)}{p(y\mid x)}}\right]$
Le Cam	$\frac{1-x}{2x+2}$	$\mathbb{E}_{p(x,y)}\frac{[p(y\mid x)-p(y)]^2}{2p(y\mid x)+2p(y)}$
Jensen Shannon	$x\log\frac{2x}{x+1}+\log\frac{2}{x+1}$	$\mathbb{E}_{p(x,y)}\left[\log\frac{2p(y\mid x)}{p(y\mid x)+p(y)}+\frac{p(y)}{p(y\mid x)}\log\frac{2p(y)}{p(y\mid x)+p(y)}\right]$
Reverse KL	$-\log x$	$\mathbb{E}_{p(x,y)}\left[\frac{p(y)}{p(y\mid x)}\log\frac{p(y)}{p(y\mid x)}\right]$

B. Discriminator Noise Visualization

Besides the visualization of the discriminator noise when the policy is trained using the logarithmic reward (see Fig. 6), we also visualize the discriminator noise when using the accuracy-based and the clipped linear reward in Fig. 10. We can see that the discriminator noise when using the clipped linear reward has a smaller bias and variance.



(a) Accuracy-based reward

(b) Clipped linear reward

Figure 10: Visualization of the discriminator noise of RAM trained using the accuracy-based (cf. Eq. (1)) and the clipped linear reward (cf. Eq. (17)).

C. Reward Clipping

We compare the performance of models trained using the logarithmic and the linear reward with and without reward clipping. Experimental results are that the clipped logarithmic reward achieves almost the same performance on the digit recognition task on both RAM and DT-RAM, slightly better performance on the skill discovery task (~ 1.5 more learned skills), and slightly worse performance ($\sim 3.5\%$ lower accuracy) on the object counting task. The clipped linear reward achieves almost the same performance on the object counting task, slightly better performance ($\sim 1\%$ and $\sim 1.5\%$ higher accuracy on RAM and DT-RAM respectively) on the digit recognition task, and considerable improvement (~ 23 more learned skills) on the unsupervised skill discovery task (see Fig. 11). These results suggest that reward clipping is a generally beneficial technique, which is consistent with our theoretical analysis in Sec. 4.5.

D. Evaluation of Various g Functions

We evaluate several other g functions in addition to the linear and logarithmic functions using the RAM model on the digit recognition task. Fig. 12 illustrates the clipped generalized reward with respect to the estimated posterior probability when using different g functions (cf. Eq. (9)). The reward is clipped at $q_\phi(y \mid \tau) = p(y) = 0.1$ (cf. Eq. (17)). Fig. 13 shows training curves when using different g functions. We can see that the linear function results in the best performance, and

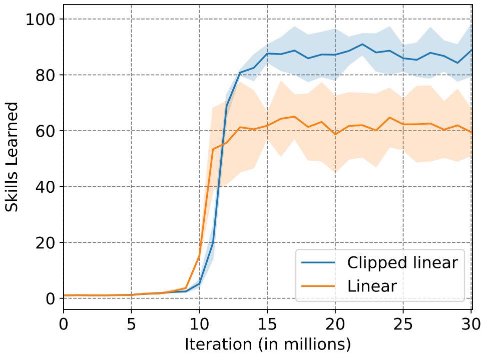


Figure 11: Reward clipping on the unsupervised skill discovery task.

g functions that are similar in shape to the linear function generally perform well. The logarithmic function and function $g(x) = x^6$ perform worse than others, which can be explained from the perspective of the requirements of g functions. Though the logarithmic function works ideally in information transmission in theory where noise is not an issue, it suffers from noisy rewards as discussed in Sec. 4. Function $g(x) = x^6$, on the other hand, leads to a small bias and variance of the estimated reward, which suggests a favorable ability in noise moderation. However, it cannot transmit information with high fidelity. Its incompetence in information transmission can be observed from the shape of the corresponding plot in Fig. 12, where a wide range of values, e.g., [0, 0.5], is compressed to values close to zero, leading to a substantial ignorance of information in various observations. In contrast, the linear function achieves a trade-off between these two abilities and exhibits the best performance among all the g functions considered.

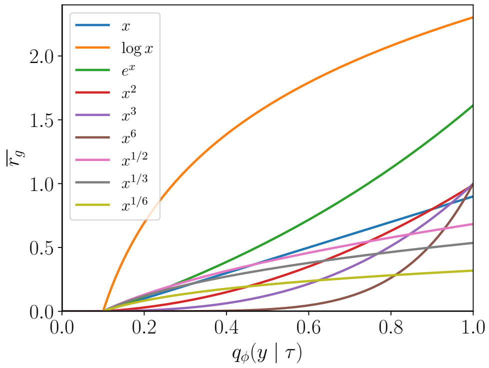


Figure 12: Clipped generalized reward with respect to the estimated posterior probability.

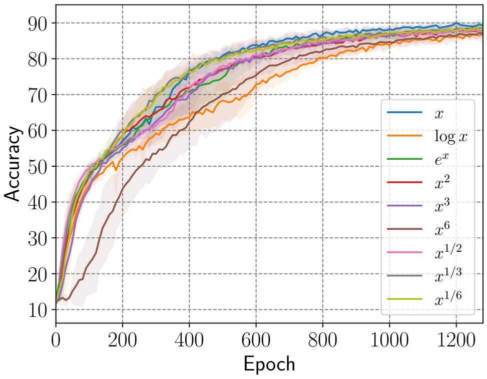


Figure 13: Evaluation of various g functions using the RAM model on the digit recognition task.

E. Environments

Cluttered-MNIST We generate the Cluttered MNIST dataset by a generator provided by the code repository[5] of Sønderby et al. (2015), where we adopt the dataset configuration from Mnih et al. (2014). A Cluttered MNIST image is generated by randomly placing an original MNIST image (28×28) and 4 randomly cropped patches (8×8) from original MNIST images in an empty image (60×60). We generate 60k Cluttered MNIST images, of which 90% are used for training and the rest for validation.

Four-room environment The four-room environment is adopted from Strouse et al. (2022) and is shown in Fig. 3b. There are four rooms and 104 states. The agent is initialized at the top-left corner at each episode and can select an action from $\{left, right, up, down, no\text{-}op\}$ at each time step. The length of each trajectory is 20, by which the agent is able to reach all

[5] https://github.com/skaae/recurrent-spatial-transformer-code

but one state, arising the maximum number of possible learned skills 103. The target skill label is uniformly sampled as an integer in $[0, 127]$ at each episode.

Object counting We create a simulation environment for the task of object counting in occlusion based on the simulation environment provided by the code repository[8] of Li et al. (2021). We use cubes of three different sizes (*small*, *medium*, and *large*) in two different colors (*red*, and *blue*) as objects on the table. The goal object is one of the *small* or *medium* objects. Each scene is initialized under the following constraints: 1) at least one large object is on the table as an abstraction; 2) the number of other objects N is sampled from a Poisson distribution ($\lambda = 4$) and is clipped at a maximum number of 6; 3) one of the goal objects is occluded by an object of a larger size with a probability of 80% to make occlusion happen frequently. The number of goal objects is uniformly sampled between 0 and N. The agent is initialized in front of the table and takes as input an egocentric RGB image with a resolution of 256×256 (cf. Fig. 3c). The agent has three discrete actions: *rotate_right*, *rotate_left*, and *stop*. The agent circles around the table by 30 degrees with each rotation action. The maximum number of movement steps is 6, by which the agent can move to the opposite of its initial position. We generate offline datasets for training (100k scenes) and evaluation (1k scenes) because online occlusion checking including scene initialization in the CoppeliaSim simulator is slow.

F. Implementation

RAM We use an existing implementation of the original RAM model[6]. Given an image and the coordinate of the glimpse, a glimpse network extracts visual representations of the attended patch by an MLP. The coordinate is mapped into representations by another MLP. The two representation vectors have the same dimensionality of 256. They are added together to get the glimpse representations. A simple RNN as the core network recurrently processes glimpse representations and produces hidden representations with a dimensionality of 256 at each time step. A policy network takes hidden representations of the core network as input to predict the location of the next glimpse. When the maximum number of movement steps is reached, a classification network takes hidden representations of the core network as input to produce the class prediction and finalize the task. The maximum number of movement steps is 18 in our experiments. The original RAM uses multi-resolution glimpses at each time step for achieving higher classification accuracy. The glimpse of the lowest resolution can cover almost the entire image. This setting compromises the quality of the attention policy. To focus on policy learning in this work, we use a single small glimpse of size 4×4 at each time step. The idea of not using multi-resolution glimpses has been used by Elsayed et al. (2019) for better interpretability. In our experiments, RAM models are trained using REINFORCE (Williams, 1992) and optimized by Adam (Kingma & Ba, 2015) for 1500 epochs with a batch size of 128 and a learning rate of 3e-4.

DT-RAM The DT-RAM model used in the experiments is from our own implementation. Instead of using two separate policy networks for location prediction and task termination respectively, which is designed for curriculum learning in the original DT-RAM, we use an integrated policy network for both location prediction and task termination. Same as RAM, the glimpse size is 4×4, and the maximum number of movement steps is 18 for DT-RAM. In our experiments, DT-RAM models are trained for 1500 epochs with the same optimization configuration as RAM models.

Model for unsupervised skill discovery The implementation of the model for unsupervised skill discovery is based on the code repository[7] of Strouse et al. (2022). In this implementation, the model uses the last state as an abstraction of the trajectory. The model is trained using a distributed actor-learner setup similar to R2D2 (Kapturowski et al., 2019). The Q-value targets are computed with Peng's $Q(\lambda)$ (Peng & Williams, 1996) instead of n-step double Q-learning. Following Strouse et al. (2022), performance of the agent is evaluated using the number of learned skills

$$n_{\text{skills}} = 2^{\mathbb{E}[\log q_\phi(y|\tau) - \log p(y)]}, \tag{20}$$

which can be understood as the measurement of the logarithmic reward in bits.

Model for object counting The implementation of the model for robotic object counting is based on the code repository[8] of Li et al. (2021). We replace the REINFORCE algorithm with PPO for more efficient training. The implementation of the

[6] https://github.com/kevinzakka/recurrent-visual-attention

[7] https://github.com/deepmind/disdain

[8] https://github.com/mengdi-li/robotic-occlusion-reasoning

PPO algorithm is based on the code repository[9] of Chevalier-Boisvert et al. (2019). The model consists of a pretrained and fixed ResNet18 (He et al., 2016) to extract feature maps from its *conv3* layer. The feature maps are then passed through two CNN layers and an average pooling layer to get visual representations of dimension 256. The index of the target object is mapped into a 10-dimensional embedding, which is called the goal representation. The visual and goal representations are concatenated together as the input of an RNN network, which recurrently produces hidden representations at each time step for the policy network and classification network. When the policy network selects the *stop* action, the classification network is triggered to produce the prediction of the number of the target object. We train the model for 2M episodes. Five processes are used to collect experience with a horizon of 40 steps. We train the model using Adam (Kingma & Ba, 2015) with a learning rate of 1e-4. Other hyperparameters of PPO are the same as the original implementation[9] except that we use 10 epochs of minibatch optimization and 5 parallelization processes.

G. DISDAIN

The reward of the DISDAIN method is $r = r_{\text{log}} + \lambda r_{\text{DISDAIN}}$, where r_{log} is the logarithmic reward function (cf. Eq. (4)), λ is a weighting coefficient, and r_{DISDAIN} is an auxiliary ensemble-based reward calculated as

$$r_{\text{DISDAIN}} = \mathbb{H}\left[\frac{1}{N}\sum_{i=1}^{N} q_{\phi_i}(y \mid \tau)\right] - \frac{1}{N}\sum_{i=1}^{N} \mathbb{H}\left[q_{\phi_i}(y \mid \tau)\right], \tag{21}$$

where N is the number of discriminators of the ensemble, and $\mathbb{H}[X]$ is the entropy of random variable X. The DISDAIN reward is essentially the estimation of the epistemic uncertainty of the discriminator.

H. Additional Results

H.1. Case Study

H.1.1. Hard attention for digit recognition

In Fig. 14, we provide cases of the DT-RAM model on the digit recognition task for intuitive comparison between the model trained using different reward functions. All the cases are randomly sampled without any cherry-picking. We can see that trajectories generated by the model trained using the clipped linear reward can cover sufficient information for recognizing the digit, while trajectories generated by the model trained using the logarithmic reward function tend to be pessimistic, e.g., trajectories in cases of digit 9 and digit 6 in the first row, digit 0 in the second row, and digit 4 in the third row. The exploration trajectories generated by the model trained using the accuracy-based reward tend to sample less informative areas, e.g., trajectories in cases of digit 6 in the first row, and digit 2 in the third row and second column, which may account for its low accuracy.

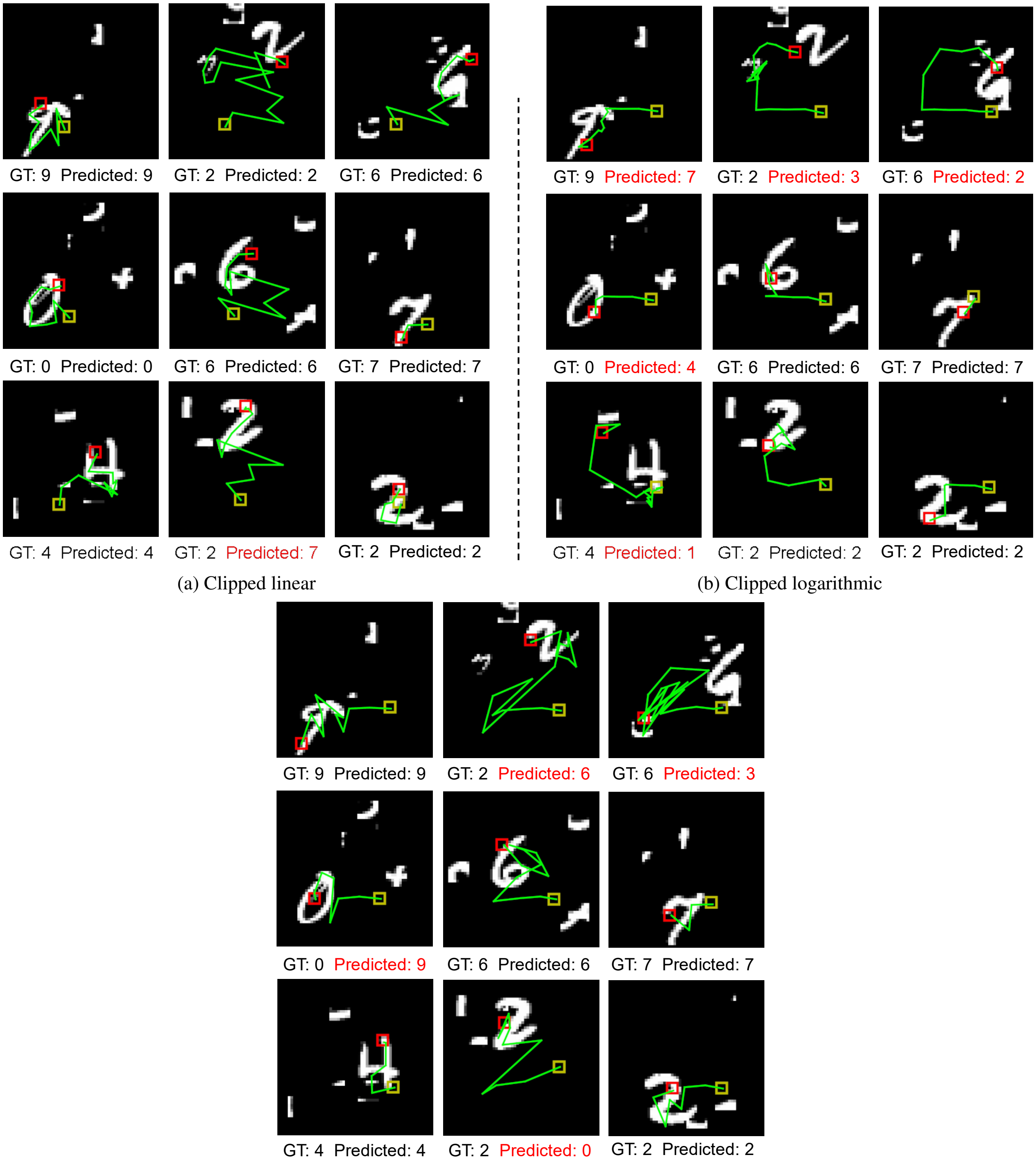
H.1.2. Robotic object counting

Fig. 15 shows examples of the pessimistic exploration issue when using the logarithmic and the accuracy-based reward function. The agent trained using the accuracy-based reward function chooses not to move, and the agent trained using the logarithmic reward function terminates exploration too early to acquire sufficient information for predicting the number of the target object. They guess the number of target objects based on insufficient observations, while the agent trained using the clipped linear reward function learns to choose a reasonable number of movement steps to explore the environment.

H.2. State Occupancy in Unsupervised Skill Discovery

Fig. 16 demonstrates state occupancy reached using different reward functions at initialization, at the intermediate stage, and at convergence during training. We can see that using the clipped linear reward function, the agent learns to reach all states as using the DISDAIN reward, while the agent mainly explores the first room when using the clipped logarithmic reward function.

[9] https://github.com/mila-iqia/babyai



(a) Clipped linear

(b) Clipped logarithmic

(c) Accuracy-based

Figure 14: Comparison of DT-RAM models trained by different reward functions. GT: the ground-truth class; Predicted: the predicted class. Red indicates incorrect predictions.

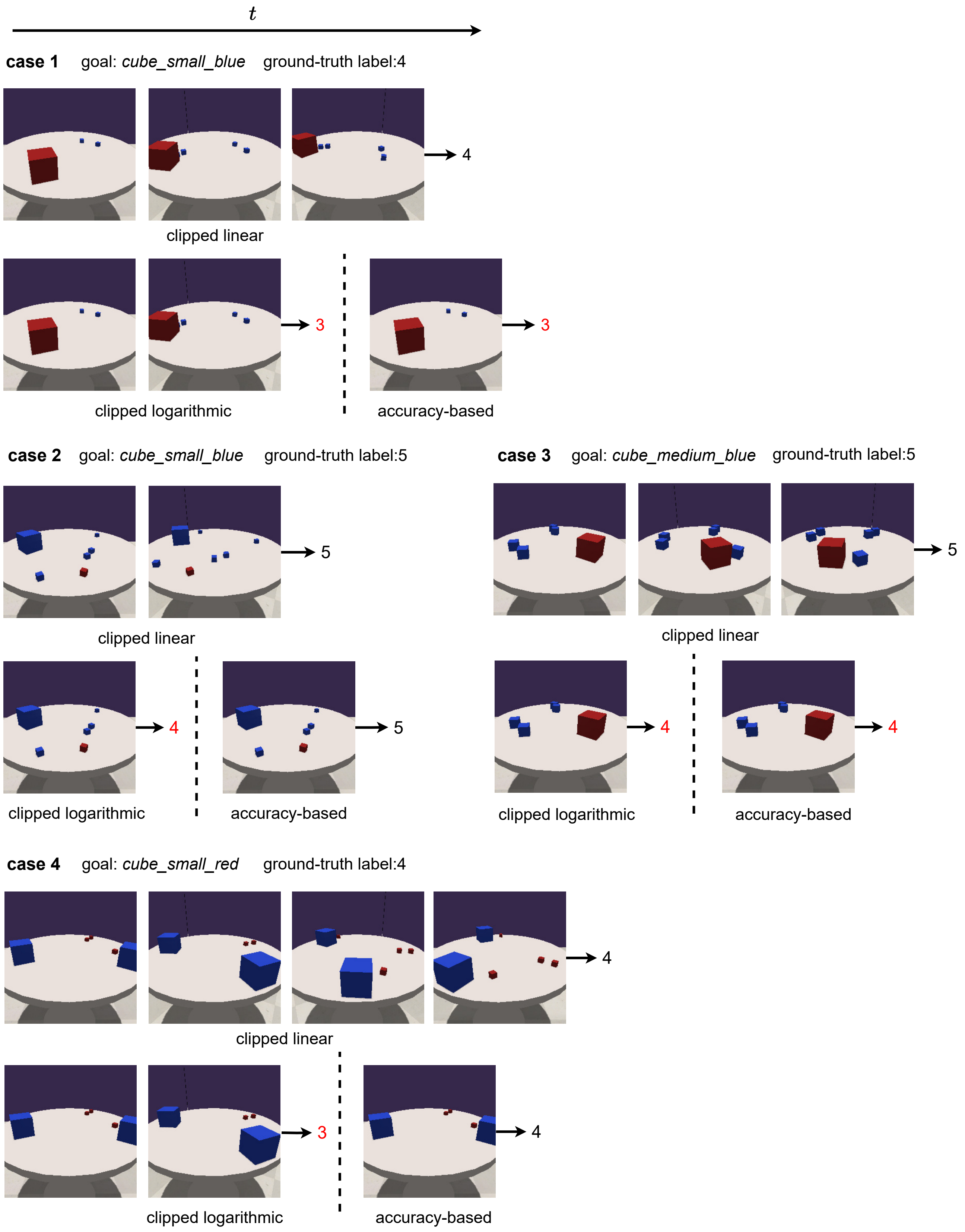
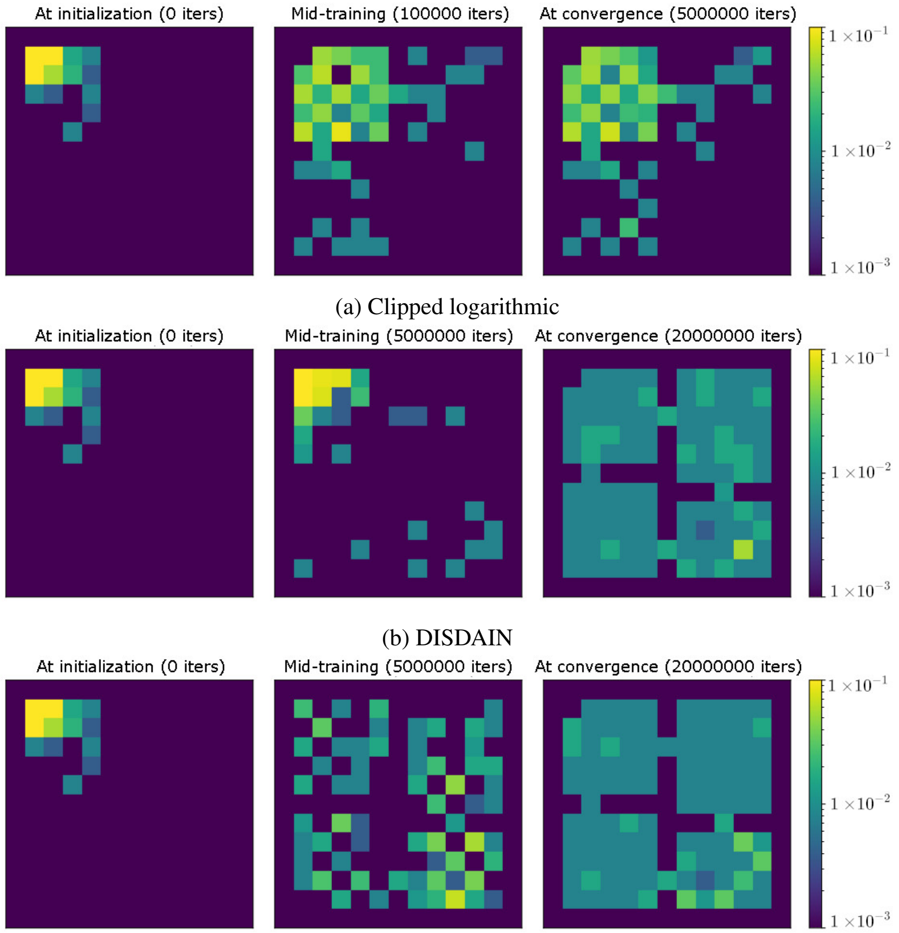

t
case 1 goal: *cube_small_blue* ground-truth label:4
4
clipped linear
3
3
clipped logarithmic
accuracy-based
case 2 goal: *cube_small_blue* ground-truth label:5
5
clipped linear
4
5
clipped logarithmic
accuracy-based
case 3 goal: *cube_medium_blue* ground-truth label:5
5
clipped linear
4
4
clipped logarithmic
accuracy-based
case 4 goal: *cube_small_red* ground-truth label:4
4
clipped linear
3
4
clipped logarithmic
accuracy-based

Figure 15: Comparison of models trained by different reward functions on the robotic object counting task. The number next to the arrow after a sequence of egocentric views is the number of goal objects predicted by the agent. Red numbers indicate wrong predictions.



(a) Clipped logarithmic

(b) DISDAIN

(c) Clipped linear

Figure 16: States reached using different reward functions. Plots depict ratios of final states reached after performing 10 trajectories per skill. The ratio is clipped between 0.001 and 0.1 for the sake of visualization. Note that the number of iterations at convergence is different (see Fig. 8a).