Sequential Generative Exploration Model for Partially Observable Reinforcement Learning

Haiyan Yin[1], **Jianda Chen**[1], **Sinno Jialin Pan**[1], **Sebastian Tschiatschek**[2]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Faculty of Computer Science, University of Vienna, Vienna, Austria
yinhaiyan@outlook.com, {jianda001, sinnopan}@ntu.edu.sg, sebastian@tschiatschek.net

Abstract

Many challenging partially observable reinforcement learning problems have sparse rewards and most existing model-free algorithms struggle with such reward sparsity. In this paper, we propose a novel reward shaping approach to infer the intrinsic rewards for the agent from a sequential generative model. Specifically, the sequential generative model processes a sequence of partial observations and actions from the agent's historical transitions to compile a belief state for performing forward dynamics prediction. Then we utilize the error of the dynamics prediction task to infer the intrinsic rewards for the agent. Our proposed method is able to derive intrinsic rewards that could better reflect the agent's surprise or curiosity over its ground-truth state by taking a sequential inference procedure. Furthermore, we formulate the inference procedure for dynamics prediction as a multi-step forward prediction task, where the time abstraction that has been incorporated could effectively help to increase the expressiveness of the intrinsic reward signals. To evaluate our method, we conduct extensive experiments on challenging 3D navigation tasks in ViZDoom and DeepMind Lab. Empirical evaluation results show that our proposed exploration method could lead to significantly faster convergence than various state-of-the-art exploration approaches in the testified navigation domains.

Introduction

Reinforcement learning is a formalism for autonomous agents to learn meaningful task skills through the execution of exploration-exploitation, driven by the reward signals issued by the environment (Sutton and Barto 1998). When such reward signals are sparse, it would be difficult for the agent to progress in grasping meaningful task skills. Unfortunately, many reinforcement learning problems come with sparse rewards, such as navigation, robotics control and video games playing. For instance, in many navigation domains, the agent only receives a single positive reward upon reaching the target location. Thus the agents trained under such reward sparsity would easily get stuck into a local state space, such as bumping into a wall, and thus become unable to make efficient progress for policy learning.

One inherent reason that leads to the struggle of existing algorithms with such reward sparsity is that initially, agents trained with those approaches could hardly stumble into a reward/goal state by chance, when executing their simple exploitation-exploration strategies (Pathak et al. 2017). Therefore, it is crucial to develop an efficient exploration mechanism to encourage the agent to continuously search through the state space in seek of reward gaining experience. From the existing reinforcement learning literature, one prominent line of solutions for promoting the exploration behaviors of the agent is via *reward shaping* (Singh 1992; Dorigo and Colombetti 1994; Barto, Mirolli, and Baldassarre 2013). Inspired by the animal learning theory, the reward shaping approaches mostly put the effort of identifying observations that are novel or surprising as the key factor that drives efficient learning (Barto, Mirolli, and Baldassarre 2013). As such, many of the existing works focus on developing some additional reward model to generate intrinsic reward signals as a measure of the novelty or surprise of the agent over a state (Schmidhuber 1991; Singh, Barto, and Chentanez 2004; Oudeyer, Kaplan, and Hafner 2007).

Since many of the well-adopted exploration methods from the conventional reinforcement learning literature, such as the UCB-type algorithms (Lai and Robbins 1985; Garivier and Cappé 2011), work well with tractable MDPs but are not straightforwardly applicable to deal with the tasks with high dimensional state space, it is a non-trivial task to develop an effective reward shaping method for deep reinforcement learning problems. When considering the problems under a partially observable setting, the challenge of developing an effective intrinsic reward system becomes even more severe, as it is extremely difficult to infer the novelty of the agent over its true MDP state given only partial observations. In recent years, there are a number of reward shaping exploration approaches emerged for deep reinforcement learning, which have brought significant performance improvement over many popular benchmark tasks (Pathak et al. 2017; Savinov et al. 2019; Pathak, Gandhi, and Gupta 2019). Despite their success, there are two main limitations for such methods to effectively work on tasks with partial observability. First, most of the well-adopted exploration models, such as the Intrinsic Curiosity Module (ICM) (Pathak et al. 2017) and Random Network Distillation (RND) (Burda et al. 2019) develop their intrinsic reward systems upon the local observations. However, only considering the local observations is clearly insufficient to infer the novelty over the true world

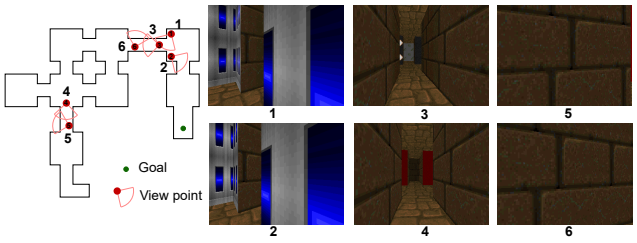

Figure 1: Partial observations derived from the *my_way_home* scenario in *ViZDoom*. The frames in each column correspond to a pair of partially observed states with visually similar partial observations but essentially different MDP states (i.e., the true MDP state would correspond to the actual coordinates of the agent's map location).

state in tasks with partial observability. For instance, in the navigation scenario shown in Figure 1, the local observations would look no different from each other at many places in the map even though their underlying true world states are essentially different. Therefore, it is crucial to develop a reward shaping method with a sequential inference procedure for the partially observable domains, so that the novelty or surprise of the agent could be inferred with a greater amount of evidence. Second, many well-adopted exploration models infer intrinsic rewards from the error of some simple downstream prediction tasks, such as self-prediction or one-step look-ahead, which might restrict the expressiveness of their inferred intrinsic reward scores. The reason is that in such self or one-step dynamics prediction tasks, the input and output states often convey great similarity, which makes those works sometimes fail to derive an expressive novelty measure to truly distinguish the novel states from the state space experienced by the agent.

In this paper, we propose a novel *Sequential Generative Exploration Model* (SGEM) to overcome the aforementioned limitations of the conventional reward shaping approaches in deep reinforcement learning domains. Overall, SGEM has the following three distinguished properties compared to most of the existing exploration approaches. First, SGEM aims at inferring the intrinsic reward signals from the novelty or surprise of the agent over its true world state. Therefore, SGEM adopts a sequential inference procedure which could effectively synthesize the past transitions for the agent to infer its intrinsic reward. Second, SGEM incorporates a multi-step dynamics prediction task to infer the intrinsic reward. By taking into account of the time abstraction during the forward dynamics prediction task, the method scales up the difficulty of the forward dynamics prediction task and thus helps the agent to derive more expressive intrinsic reward signals. Third, SGEM is general in its form and could be applied to most partially observable policy learning tasks that come with a high dimensional state space. Moreover, it consists of modeling flexibility to certain extent such that some of its modules could concurrently work with other exploration models, e.g., we integrate a RND module into our proposed generative model, which serves as a target projection function to obtain a compact state representation.

Related Work

Curiosity-driven exploration has been studied extensively in the reinforcement learning literature (Oudeyer and Kaplan 2007; Oudeyer, Kaplan, and Hafner 2007). In recent years, research on intrinsic exploration for deep reinforcement learning has developed various different measures to model the agent's curiosity, such as counts (Choi et al. 2019; Tang et al. 2017), pseudo-counts (Bellemare et al. 2016; Ostrovski et al. 2017), prediction-error (Stadie, Levine, and Abbeel 2015; Achiam and Sastry 2017; Yu, Lyu, and Tsang 2020) and information gain (Houthooft et al. 2016; Nikolov et al. 2019). For the tasks with partial observability, one prominent line of curiosity-driven exploration methods fall to the prediction-error-based category. Pathak et al. (2017) propose a forward-backward dynamics model and use the prediction loss of the forward model to infer the state curiosity. Oh and Cavallaro (2019) introduce a triplet ranking loss to push the prediction output of the forward dynamics model to be far from the output generated by taking some alternative actions. Apart from such prediction-error-based approaches, recently, Savinov et al. (2019) also introduce an associative memory-based approach. The method forms a memory of novel states and trains a comparator network to model the reachability between visited states to the novel states to compute the intrinsic reward based on the reachability score. Despite their effectiveness, all the aforementioned approaches do not explicitly consider to develop the intrinsic reward model for agents with partial observability in a sequential manner and therefore such methods could hardly infer the novelty of agent over its true world state. Our proposed exploration method leverages a sequential inference procedure to infer the curiosity of agent over a state that is closer to its true world state. To this end, it sequentially processes the historical transitions in an episode to infer the intrinsic reward of a state.

Our work is also related to the works on sequential dynamics models for reinforcement learning. In (Oh et al. 2015; Chiappa et al. 2017; Hafner et al. 2019), recurrent dynamics models are introduced to generate realistic future states for model-based planning. In (Ke et al. 2019), a sequential dynamics model is trained to generate future states and actions for model-based control. In (Ha and Schmidhuber 2018; Gregor et al. 2019), VAE models are utilized to learn high-level representations to be used as the input for policy learning in partially observable domains. In (Lee et al. 2020), a latent soft actor-critic algorithm is proposed which combines the process of representation learning with policy training. While most of the aforementioned works focus on deriving realistic future predictions from the dynamics model, our work performs dynamics prediction with the sequential generative model for curiosity-driven exploration. Furthermore, most of the existing dynamics models formulate the task as one-step look-ahead or unsupervised prediction, whereas we consider multi-step forward dynamics prediction to derive more expressive intrinsic rewards. The work from Gregor et al. (2019) also introduces a multi-step dynamics prediction loss when training a sequential-VAE model. However, they focus on the task of unsupervised learning and their method works with a different inference structure from ours.

Background

Partially Observable Markov Decision Processes (POMDPs) generalize MDPs by learning under partial observability. Formally, a POMDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{Z}, \mathcal{R} \rangle$, where $\mathcal{S}$, $\mathcal{O}$ and $\mathcal{A}$ define the state space, the observation space and the action space, respectively. At each step, the agent receives an observation $\mathbf{o}_t \in \mathcal{O}$ and determines an action to take based on a policy function $\pi(\cdot)$. The transitions of the POMDP are defined on the state space of the underlying MDP: $\mathcal{T}(\mathbf{s}, \mathbf{a}, \mathbf{s}') = p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ such that $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ is the probability of transiting to state $\mathbf{s}'$ after taking action $\mathbf{a}$ in state $\mathbf{s}$. $\mathcal{Z}(\mathbf{s}, \mathbf{a}, \mathbf{o}) = p(\mathbf{o}|\mathbf{s}, \mathbf{a})$ specifies the probability of receiving observation $\mathbf{o}$ when taking action $\mathbf{a}$ in state $\mathbf{s}$. The reward function $\mathcal{R}(\mathbf{s}, \mathbf{a})$ defines the real-valued environment reward obtained by the agent when taking action $\mathbf{a}$ in state $\mathbf{s}$. Under partial observability, the state space $\mathcal{S}$ is not directly accessible by the agent and the agent performs decision making by forming a belief state $\mathbf{b}_t$ which is updated upon receiving new observations or rewards. In this work, we consider inferring the belief state by filtering the entire past partial observations of the agent, i.e., $(\mathbf{o}_0, \mathbf{o}_1, ..., \mathbf{o}_t)$. The goal of reinforcement learning is to optimize a policy $\pi(\mathbf{b}_t)$ which outputs an action distribution given each *belief* state $\mathbf{b}_t$, with the objective of maximizing the discounted cumulative rewards collected from each episode, i.e., $\sum_{t=0}^{\infty} \gamma^t r_t$, where $\gamma \in [0, 1)$ is a real-valued discount factor.

Intrinsic Exploration Framework

In this paper, our primary focus is on tasks with partial observations where the observations are defined as high-dimensional inputs (i.e., images) and the external rewards r_t are sparse, i.e., zero for most of the time. To tackle such challenge, we define a general intrinsic reward function from SGEM to evaluate the novelty or surprise of the agent over its experience in the world, so as to issue a step-wise exploration reward bonus to the agent. We illustrate the main components for SGEM in Figure 2. The exploration framework consists of three main components: an inference network, an observation decoder and a target model which we define in the form of random network distillation (RND). Each of the components is modeled as a neural network-based function.

We formulate a multi-step dynamics prediction task with SGEM to derive an expressive intrinsic reward measure. Specifically, to generate a state for step $t + \delta$, the inference model for SGEM filters over the past transition sequence composed by a state observation sequence $\mathbf{o}_{1:t}$ and an action sequence $\mathbf{a}_{t:t+\delta-1}$ to generate a latent state $\mathbf{z}_{t+\delta}$, where $\mathbf{o}_{1:t} = (\mathbf{o}_1, ..., \mathbf{o}_t)$ and $\mathbf{a}_{t:t+\delta-1} = (\mathbf{a}_t, ..., \mathbf{a}_{t+\delta-1})$ represent a sequence of partial observations and actions, respectively. SGEM is essentially performing a multi-step forward dynamics prediction task to generate latent state $\mathbf{z}_{t+\delta}$, considering that the inference model filters the observation sequence only up to time t and the source of information for task dynamics over the subsequent time span of δ comes from the action sequence only. As such, the inference structure for our proposed model is different from those conventional sequential state space models (e.g., (Gregor et al. 2019; Hafner et al. 2019; Ke et al. 2019)), since their inference models are mostly defined over one-step look-ahead with an autoregressive nature.

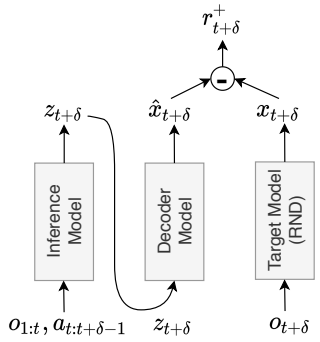

Figure 2: An overview to the procedure of inferring intrinsic reward bonus with SGEM.

Given the latent state $\mathbf{z}_{t+\delta}$, the decoder model for SGEM decodes a state for time $t + \delta$. Rather than generating the observation at its original high dimensional representation space (i.e., image pixels), we utilize RND to form an effective target embedding for the observations, which projects the high dimensional observation $\mathbf{o}_{t+\delta}$ into an embedding space characterized by some random projection functions. By utilizing RND, we could reduce considerable amount of computational cost since the size of target embedding derived from RND is often much smaller than the original high dimensional observations. At the same time, our method leverages the advantage of RND to derive an effective novelty measure which quantifies the novelty reward as the uncertainty of distilling a randomly drawn function from its prior (Osband, Aslanides, and Cassirer 2018).

Sequential Generative Model

Generally, sequential generative models are hard to train and therefore we resort to variational inference. Overall, our task of interest is to learn a distribution to generate a sequence of observation embeddings $(\mathbf{x}_{\delta+1}, ..., \mathbf{x}_{T+\delta})$, given the partial observations $\mathbf{o}_{1:T}$ and actions $\mathbf{a}_{1:T+\delta-1}$, where $\mathbf{x}_t$ denotes the target embedding for $\mathbf{o}_t$. To generate $\mathbf{x}_{\delta+1:T+\delta}$, we consider the following probabilistic model:

$$p_\theta(\mathbf{x}_{\delta+1:T+\delta}, \mathbf{z}_{\delta+1:T+\delta}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t+\delta}|\mathbf{z}_{t+\delta})\, p(\mathbf{z}_{t+\delta}),$$

where $\mathbf{z}_{\delta+1:T+\delta}$ denote a set of latent states for generating the observation embeddings from step $t = \delta + 1$ to $T + \delta$, $p(\mathbf{z}_{t+\delta})$ is the prior distribution and θ is the parameter for the decoder. For simplicity we assume $\mathbf{z}_{\leq\delta} = 0$. All the conditional distributions and prior distributions are represented as simple distributions, e.g., Gaussian distributions. Even though each single distribution is uni-modal, marginalizing over all the sequence of latent variables would make them highly multi-modal, and therefore suffice to model the latent distribution in our target applications.

To generate $\mathbf{x}_{\delta+1:T+\delta}$, SGEM learns a latent state distribution. We define a posterior distribution $q_\phi(\cdot)$ parameterized

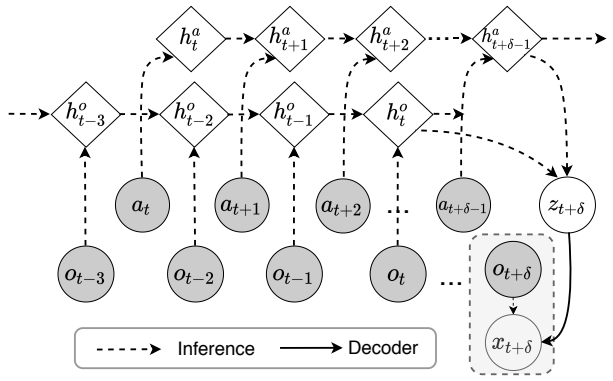

Figure 3: The backbone for the inference model and the decoder model in SGEM. The inference model performs a multi-step dynamics prediction task to generate a latent code $\mathbf{z}_{t+\delta}$ given the observation sequence $\mathbf{o}_{1:t}$ and an action sequence $\mathbf{a}_{t:t+\delta-1}$. Then the generative model decodes the latent state to predict the future observation, where the target for prediction is modeled as the random state embedding projected by RND. For ease of understanding, we only show the inference and generative procedure for a single state. The model could be straightforwardly unrolled to generate the following states.

by ϕ as follows,

$$q_\phi(\mathbf{z}_{\delta+1:T+\delta}|\mathbf{o}_{1:T},\, \mathbf{a}_{1:T+\delta-1}) = \prod_{t=1}^{T} q_\phi(\mathbf{z}_{t+\delta}|\mathbf{o}_{1:t},\, \mathbf{a}_{t:t+\delta-1}).$$

As such the posterior distribution for $\mathbf{z}_{t+\delta}$ is derived by filtering over the observations sequence $\mathbf{o}_{1:t}$ and the action sequence $\mathbf{a}_{t:t+\delta-1}$. In practice, the inference network for SGEM adopts two aggregation functions $f_h^o(\cdot)$ and $f_h^a(\cdot)$ to process the observations and actions in sequence, to derive synthesized features, i.e., $\mathbf{h}_t^o$ and $\mathbf{h}_{t+\delta-1}^a$, for representing the observation sequence and the action sequence (as illustrated in Figure 3). The computation of $\mathbf{z}_{t+\delta}$ would depend on $\mathbf{h}_t^o$ and $\mathbf{h}_{t+\delta-1}^a$.

To train our proposed sequential generative model with the approximate posterior, we derive the Evidence Lower Bound (ELBO) as follows:

$$\begin{aligned}
&\log p_\theta(\mathbf{x}_{\delta+1:T+\delta},\, \mathbf{z}_{\delta+1:T+\delta}) \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}_{\delta+1:T+\delta}|\mathbf{o}_{1:T},\mathbf{a}_{1:T+\delta-1})}\Big[\frac{\log p_\theta(\mathbf{x}_{\delta+1:T+\delta},\, \mathbf{z}_{\delta+1:T+\delta})}{\log q_\phi(\mathbf{z}_{\delta+1:T+\delta}|\mathbf{o}_{1:T},\, \mathbf{a}_{1:T+\delta-1})}\Big] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}_{\delta+1:T+\delta}|\mathbf{o}_{1:T},\mathbf{a}_{1:T+\delta-1})}\Big[\log p_\theta(\mathbf{x}_{\delta+1:T+\delta},\, \mathbf{z}_{\delta+1:T+\delta})\Big] \\
&\quad - \mathbb{KL}\left(q_\phi(\mathbf{z}_{\delta+1:T+\delta}\,|\,\mathbf{o}_{1:T},\, \mathbf{a}_{1:T+\delta-1})\,\|\,p\left(\mathbf{z}_{\delta+1:T+\delta}\right)\right)
\end{aligned}$$

By leveraging the temporal structure of the inference network and the decoder, we further break down the ELBO in the following manner to form the loss $\mathcal{L}_{SGEM}$ for optimizing our proposed SGEM model:

$$\begin{aligned}
\mathcal{L}_{SGEM} = -\sum_{t=1}^{T} &\mathbb{E}_{q_\phi(\mathbf{z}_{t+\delta}|\mathbf{o}_{1:t},\mathbf{a}_{t:t+\delta-1})}\log p_\theta(\mathbf{x}_{t+\delta}|\mathbf{z}_{t+\delta}) \\
&+ \beta\,\mathbb{KL}\big(q_\phi(\mathbf{z}_{t+\delta}|\mathbf{o}_{1:t},\mathbf{a}_{t:t+\delta-1})\,\|\,p\left(\mathbf{z}_{t+\delta}\right)\big),
\end{aligned}$$

Algorithm 1 Policy Training with SGEM

Input: learning rate $\alpha > 0$, hyperparameters λ, β, δ
Initialize RL policy parameters ω, SGEM parameters θ, ϕ and a RND function $f^*(\cdot)$.
for $e = 1$ **to** MAXITER **do**
  Reset the environment and receive $\mathbf{o}_0$.
  Set transition buffer to Ø.
  for $t = 1$ **to** TIMEOUT **do**
    Sample an action $\mathbf{a}_t \sim \pi_\omega(\mathbf{o}_{0:t-1})$.
    Step the environment: $\mathbf{o}_t, r_t,$ term $\sim$ env$(\mathbf{a}_t)$.
    Save the transition to the buffer.
    if $t =$BUFFERSIZE or term **then**
      Compute r^+ for the states with index $> \delta$ following Eq (1).
      Update ω, θ, ϕ following Eq (2).
      Remove transitions with index $> \delta$ from buffer.
    end if
    if term **then** break
  end for
end for

where $\beta > 0$ is the weight for the KL-divergence term.

Policy Training with Intrinsic Rewards

We infer the intrinsic reward for a state from the error of performing multi-step dynamics prediction with SGEM. In practice, we formulate the intrinsic reward in the form of the MSE loss. At step t, the reward bonus or curiosity score is computed in the following manner:

$$\begin{aligned}
\mathbf{x}_t = f^*(o_t), \quad \hat{\mathbf{x}}_t = f_{\text{SGEM}}(\mathbf{o}_{1:t-\delta}, \mathbf{a}_{t-\delta:t-1}; \theta, \phi), \\
r_t^+ = \frac{\eta}{2}||\hat{\mathbf{x}}_t - \mathbf{x}_t||_2^2, \qquad (1)
\end{aligned}$$

where $\hat{\mathbf{x}}_t$ denotes the predicted embedding generated by the SGEM function $f_{\text{SGEM}}(\cdot)$, $\mathbf{x}_t$ denotes the target embedding generated by the RND function $f^*(\cdot)$ and r_t^+ denotes the intrinsic reward. $\eta \geq 0$ is a positive weight to scale the intrinsic reward.

The intrinsic reward module can be trained simultaneously with the reinforcement learning objective. The compound objective function for training reinforcement learning policy with our proposed curiosity-driven exploration becomes:

$$\max_{\theta,\phi,\omega} \mathbb{E}_{\pi_\omega(\mathbf{o}_t^o;\,\omega)}\left[\sum_t (r_t + r_t^+) - \lambda\mathcal{L}_{SGEM}\right], \qquad (2)$$

where θ, ϕ and ω are the parameters for the observation decoder, posterior inference model and the policy model respectively, and $\lambda \geq 0$ is a weight to balance the objectives for reward maximization and that for training SGEM. The policy model and SGEM could partially share the observation embedding parameters. The complete algorithm for policy training with SGEM is shown in Algorithm 1.

Implementation with a Dual-LSTM Architecture

In order to derive high-quality features to summarize the observation and action sequences, we present an effective

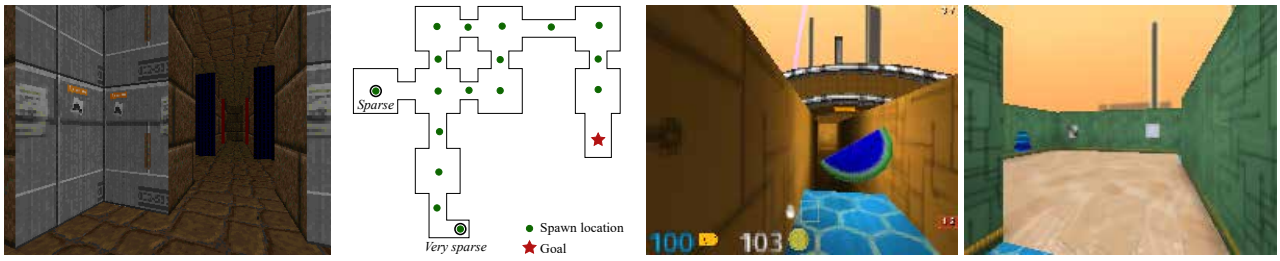

Figure 4: The 3D navigation task domains for empirical evaluation: (1) an example of partially observable frame from *ViZDoom*; (2) the specifications of spawn/goal location for the *ViZDoom* scenarios; (3/4) an example of partially observable frame from the *apple-distractions/goal-exploration* task in *DeepMind Lab*.

way to implement the posterior inference network in SGEM with a dual-LSTM architecture. To predict the latent state $\mathbf{z}_t$ for step t, we denote the input observation sequence and action sequence to the inference network as $\mathbf{O}_t = \mathbf{o}_{1:t-\delta}$ and $\mathbf{A}_t = a_{t-\delta:t-\delta-1}$, respectively. Specifically, each partial observation $\mathbf{o}_t$ is represented as a 3D image frame with width m, height n and channel c, i.e., $\mathbf{o}_t \in \mathbb{R}^{m\times n\times c}$. Each action is modeled as a 1-hot encoding vector $a_t \in \mathbb{R}^{|A|}$, where $|A|$ denotes the size of the action space. Given the sequences $\mathbf{O}_t$ and $\mathbf{A}_t$, the inference network first adopts an embedding module $f_e(\cdot)$ parameterized by $\theta_e = \{\theta_e^o, \theta_e^a\}$ to process each observation and action in the sequences as follows,

$$\boldsymbol{\phi}_t^{\mathbf{O}} = f_e(\mathbf{O}_t; \theta_{E_o}) \text{ and } \boldsymbol{\phi}_t^{\mathbf{A}} = f_e(\mathbf{A}_t; \theta_{E_a}), \tag{3}$$

where θ_e^o and θ_e^a denote the parameters for the observation embedding function and the action embedding function. Next, we adopt LSTM as the aggregation function to synthesize both sequences and generate synthesized observation/action embeddings,

$$\begin{aligned}[\mathbf{h}_t^o, \mathbf{c}_t^o] &= \text{LSTM}_o\Big(\boldsymbol{\phi}_t^{\mathbf{O}}, \mathbf{h}_{t-1}^o, \mathbf{c}_{t-1}^o\Big),\\ [\mathbf{h}_t^a, \mathbf{c}_t^a] &= \text{LSTM}_a\Big(\boldsymbol{\phi}_t^{\mathbf{A}}, \mathbf{h}_{t-1}^a, \mathbf{c}_{t-1}^a\Big),\end{aligned} \tag{4}$$

where $\mathbf{h}_t^o \in \mathbb{R}^l$ and $\mathbf{h}_t^a \in \mathbb{R}^l$ represent the latent features encoded from the observation sequence and action sequence. For simplicity, we assume $\mathbf{h}_t^o$ and $\mathbf{h}_t^a$ have the same dimensions. $\mathbf{c}_t^o$ and $\mathbf{c}_t^a$ denote the cell output for the two LSTM modules. A multiplicative interactive modeling is adopted to synthesize the observation sequence feature $\mathbf{h}_t^o$ and the action sequence feature $\mathbf{h}_t^a$ to derive a multiplicative latent code $\mathbf{h}_t^i$, and the belief state $\mathbf{b}_t$ which summarizes evidence for dynamics prediction is formed as follows:

$$\mathbf{b}_t = [\mathbf{h}_t^o, \mathbf{h}_t^a, \mathbf{h}_t^i], \text{ and } \mathbf{h}_t^i = \mathbf{h}_t^o \odot \mathbf{h}_t^a \tag{5}$$

where $\odot$ denotes element-wise multiplicative interaction. Then the latent code is sampled from a Gaussian distribution with its mean and variance generated by projecting $\mathbf{b}_t$ with a fully connected layer $f_p(\cdot)$:

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t), \quad \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t = f_p(\mathbf{b}_t). \tag{6}$$

Experiments

Experimental Setup

Task Domains For empirical evaluation, we adopt three 3D navigation tasks with first-person view: 1) '*DoomMyWayHome-v0*' from *ViZDoom* (Kempka et al. 2016); 2) '*Stairway to Melon*' from *DeepMind Lab* (Beattie et al. 2016); 3) '*Explore Goal Locations*' from *DeepMind Lab*. Specifically, '*DoomMyWayHome-v0*' allows us to test the algorithms in scenarios with varying degrees of reward sparsity, '*Stairway to Melon*' allows us to test the algorithms in scenarios with reward distractions, and '*Explore Goal Locations*' allows us to test the algorithms in scenarios with procedurally generated maze layout and random goal locations. Source code for SGEM is available in tensorflow where the details on implementation and hyperparameter settings for each task domain are also available.

Baseline Methods For fair comparison, we adopt 'LSTM-A3C' as the RL algorithm for all the methods. In the experiments, we compare with the vanilla 'LSTM-A3C' as well as the following intrinsic exploration baselines: 1) the Intrinsic Curiosity Module (Pathak et al. 2017), denoted as 'ICM'; 2) Episodic Curiosity through reachability (Savinov et al. 2019), denoted as 'EC'; 3) the Random Network Distillation model (Burda et al. 2019), denoted as 'RND'. Our proposed Sequence-level Generative Exploration Module is denoted as 'SGEM'. SGEM adopts action sequence length of 6 for the *ViZDoom* tasks and 3 for *DeepMind Lab* tasks. The baseline 'EC' needs to pretrain the comparitor model with environment transitions so we shift the corresponding learning curves by the budgets of pretraining frames (i.e., 0.6M) in the results, following the original paper (Savinov et al. 2019).

Evaluation with Varying Reward Sparsity

Our first empirical domain is a navigation task in the '*DoomMyWayHome-v0*' scenario from *ViZDoom*. The task consists of a static maze layout and a fixed goal location. At the start of each episode, the agent spawns from one of the 17 spawning locations, as shown in Figure 4. In this domain, we adopt three different setups with varying degree of reward sparsity, i.e., *dense*, *sparse*, and *very sparse*. Under the *dense* setting, the agent spawns at one randomly selected location from the 17 locations and it is relatively easy to succeed in navigation. Under the *sparse* and *very sparse* settings, the agent spawns at a fixed location far away from the goal. The environment issues a positive reward of $+1$ to the agent when reaching the goal. Otherwise, the rewards are 0. The episode

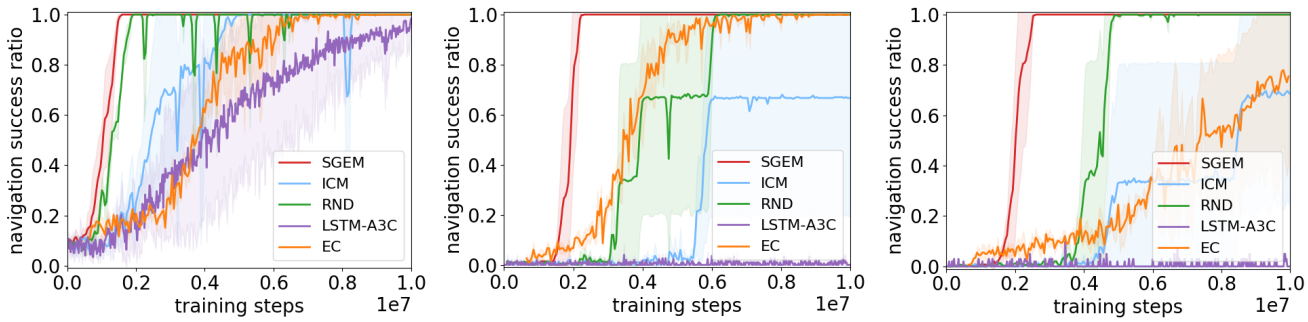

Figure 5: Learning curves measured in terms of the navigation success ratio in *ViZDoom* for the three testing scenarios (left-to-right): 1) *dense*; 2) *sparse*; 3) *very sparse*. We run each method for 6 times.

terminates when the agent reaches the goal location or the episode length exceeds the time limit of 525× 4-repeated environment steps.

We show the training curves measured in terms of navigation success ratio in Figure 5. The results from Figure 5 depicts that as the rewards go sparser, the navigation would become more challenging. The vanilla 'LSTM-A3C' algorithm could not progress at all under the *sparse* and *very sparse* settings. 'ICM' could not reach 100% success ratio under the *sparse* and *very sparse* settings, and so does 'EC' under the *very sparse* setting. Our proposed method consistently achieves 100% success ratio across all the tasks with varying reward sparsity. The detailed convergence scores are shown in Table 1.

Our proposed solution also demonstrates significant advantage in terms of convergence speed. Though the reward sparsity varies, our method could quickly reach 100% success ratio in all the scenarios. However, the convergence speeds of 'ICM', 'EC' and 'RND' apparently degrade with sparser rewards. Also, we notice that the associative memory-based method (i.e., 'EC') takes much longer time to converge compared to the prediction-error based baselines 'RND' and 'SGEM'. Moreover, 'EC' requires to pre-train the comparator module in some task domains such as *ViZDoom*, which would consume a considerable amount of pre-training data, but the other methods 'ICM' and 'RND' and 'SGEM' do not require such pre-training. Overall, our proposed method could converge to 100% success ratio on average 3.1x as fast as 'ICM' and 2.0x compared to 'RND' when measured in terms of training steps required to fully learn the task.

	dense	*sparse*	*very sparse*
LSTM-A3C	100%	0.0%	0.0%
ICM	100%	66.7%	68.6%
EC	100%	100%	75.5%
RND	100%	100%	100%
SGEM (ours)	100%	100%	100%

Table 1: Performance scores for the three maps in ViZDoom evaluated in terms of navigation success ratio upon 10m steps.

Evaluation with Varying Maze Layout and Goal Location

Our second empirical evaluation domain is a navigation task with procedurally generated maze layout and randomly chosen goal locations. We adopt the 'Explore Goal Locations' level script from *DeepMind Lab*. At the start of each episode, the agent spawns at a random location and searches for a randomly defined goal location within the time limit of 1350× 4-repeated steps. Each time the agent reaches the goal, it receives a reward of +10 and is spawned into another random location to search for the next random goal. The maze layout is procedurally generated at the start of each episode. This domain challenges the algorithms to derive general navigation behavior instead of relying on remembering the past trajectories.

We show the results with an environment interaction budget of 1.7M 4-repeated steps in Figure 6. In this task, the baseline 'EC' consumes 0.6M pretraining frames (following the setting in the released code), which makes it less feasible for the current task, as our method could obtain reasonable scores with significantly less training frames. Also note that in this challenging task with procedurally generated maze, vanilla 'LSTM-A3C' model without intrinsic curiosity exploration could only converge to an inferior performance standard of around 10. Our proposed method could score

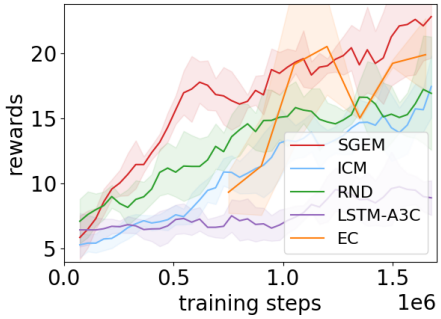

Figure 6: Learning curves for the procedurally generated goal searching task in *DeepMind Lab*.

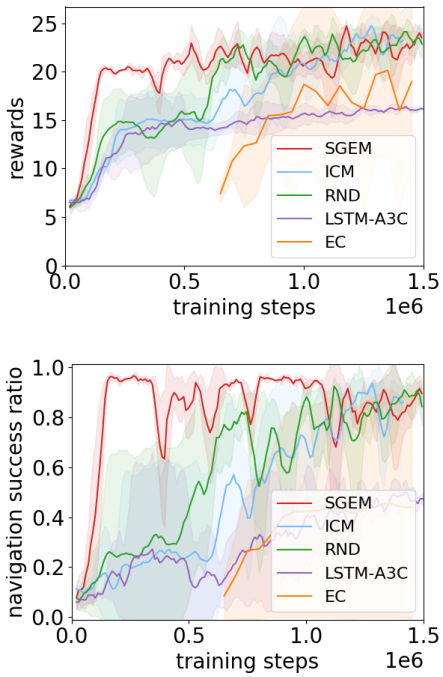

Figure 7: Learning curves for the 'Stairway to Melon' task in *DeepMind Lab*. Up: cumulative episode reward; Down: navigation success ratio.

> 20 with less than 1.3M training steps and achieve performance standard significantly higher than 'ICM' and 'RND'. This demonstrates that our proposed reward shaping method could work well in challenging partially observable tasks with procedurally changing content.

Evaluation with Reward Distractions

Our third empirical evaluation engages a cognitively complex task with reward distraction. We adopt the '*Stairway to Melon*' level script from *DeepMind Lab*. In this task, the agent can follow either two corridors: one of them leads to a dead end, but has multiple apples along the way, collecting which the agent would receive a small positive reward of $+1$; the other corridor consists of one lemon which gives the agent a negative reward of -1, but after passing the lemon, there are stairs that lead to the navigation goal location upstairs indicated by a melon. Collecting the melon makes the agent succeed in navigation and receive a reward of $+20$. The episode terminates when the agent reaches the goal location or the episode length exceeds the time limit which is specified as $525\times$ 4-repeated steps.

The results are shown in Figure 7. We demonstrate both the cumulative episode reward and the success ratio for navigation. Due to the reward distractions, the learning curves for each approach demonstrate instability with ubiquitous glitches. The vanilla 'LSTM-A3C' could only converge to an inferior navigation success ratio of $< 50\%$, and all the other baselines progress slowly. Notably, our proposed method could fast grasp the navigation behavior under the reward distraction scenario, i.e., it could achieve a navigation success ratio of $> 80\%$ with less than 0.2M environment interactions, which is at least 3x as fast as the compared baselines.

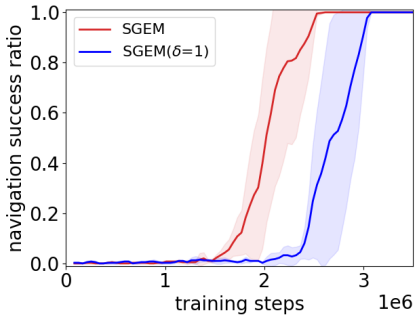

Figure 8: Results for the ablation study evaluated on the *very sparse* scenario in *ViZDoom*.

Ablation Study

We present an ablation study on the *very sparse* scenario from *ViZDoom* domain to demonstrate the privilege of performing multi-step dynamics prediction in *SGEM*. To this end, we compare our method which adopts a multi-step time span of $\delta = 6$ with a baseline method which performs one-step look-ahead, i.e., $\delta = 1$. The results are shown in Figure 8. Note that in the main results for *ViZDoom* shown in Figure 5 (c), some of the baselines considered by our work, i.e., 'RND' and 'ICM', are also performing one-step/self-prediction. From the results shown in Figure 8, we notice that performing multi-step dynamics prediction could bring noticeable privilege to the policy learning compared to the '$\delta = 1$' variant. Thus we conclude that enlarging the time span of the prediction task might help to derive more expressive intrinsic rewards and result in more efficient policy training.

Conclusions

In this paper, we tackle the challenge of improving the policy training performance in sparse rewarded partially observable domains. We propose a new exploration method which performs curiosity-driven reward shaping, termed SGEM. SGEM infers intrinsic rewards with a sequential inference network and it could derive expressive exploration rewards by considering a forward dynamics prediction task with an increased time span for prediction. In the empirical evaluation, we demonstrate our proposed method SGEM could outperform various state-of-the-art intrinsic exploration models in challenging partially observable navigation domains. Potential directions for future work include refining the prediction task being considered by SGEM to infer the intrinsic exploration reward, such as adopting an alternative inference network structure or formulating different prediction task. Also, it is worth investigating the performance of SGEM when combined with different reinforcement learning algorithms as well as when the method is applied to different application domains with partial observability.

Acknowledgements

This research is partially supported by the NTU Singapore Nanyang Assistant Professorship (NAP) grant and Singapore MOE AcRF Tier-1 grant 2018-T1-002-143 (RG131/18 (S)).

References

Achiam, J.; and Sastry, S. 2017. Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning. *CoRR* abs/1703.01732.

Barto, A.; Mirolli, M.; and Baldassarre, G. 2013. Novelty or surprise? *Frontiers in psychology* 4: 907.

Beattie, C.; Leibo, J. Z.; Teplyashin, D.; Ward, T.; Wainwright, M.; Küttler, H.; Lefrancq, A.; Green, S.; Valdés, V.; Sadik, A.; et al. 2016. Deepmind lab. *CoRR* abs/1612.03801.

Bellemare, M. G.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, NIPS 2016, Barcelona, Spain*, 1471–1479.

Burda, Y.; Edwards, H.; Storkey, A. J.; and Klimov, O. 2019. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA*.

Chiappa, S.; Racanière, S.; Wierstra, D.; and Mohamed, S. 2017. Recurrent Environment Simulators. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France*.

Choi, J.; Guo, Y.; Moczulski, M.; Oh, J.; Wu, N.; Norouzi, M.; and Lee, H. 2019. Contingency-Aware Exploration in Reinforcement Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA*.

Dorigo, M.; and Colombetti, M. 1994. Robot Shaping: Developing Autonomous Agents Through Learning. *Artificial Intelligence* 71(2): 321–370.

Garivier, A.; and Cappé, O. 2011. The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond. In *The 24th Annual Conference on Learning Theory, COLT 2011, Budapest, Hungary*, volume 19, 359–376.

Gregor, K.; Papamakarios, G.; Besse, F.; Buesing, L.; and Weber, T. 2019. Temporal Difference Variational Auto-Encoder. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA*.

Ha, D.; and Schmidhuber, J. 2018. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS 2018, Montréal, Canada*, 2455–2467.

Hafner, D.; Lillicrap, T. P.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019. Learning Latent Dynamics for Planning from Pixels. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA*, volume 97, 2555–2565.

Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; Turck, F. D.; and Abbeel, P. 2016. VIME: Variational Information Maximizing Exploration. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, NIPS 2016, Barcelona, Spain*, 1109–1117.

Ke, N. R.; Singh, A.; Touati, A.; Goyal, A.; Bengio, Y.; Parikh, D.; and Batra, D. 2019. Learning Dynamics Model in Reinforcement Learning by Incorporating the Long Term Future. *CoRR* abs/1903.01599.

Kempka, M.; Wydmuch, M.; Runc, G.; Toczek, J.; and Jaskowski, W. 2016. ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games, CIG 2016, Santorini, Greece*, 1–8.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6(1): 4–22.

Lee, A. X.; Nagabandi, A.; Abbeel, P.; and Levine, S. 2020. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS 2020, virtual*.

Nikolov, N.; Kirschner, J.; Berkenkamp, F.; and Krause, A. 2019. Information-Directed Exploration for Deep Reinforcement Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA*.

Oh, C.; and Cavallaro, A. 2019. Learning Action Representations for Self-supervised Visual Exploration. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada*, 5873–5879.

Oh, J.; Guo, X.; Lee, H.; Lewis, R. L.; and Singh, S. P. 2015. Action-Conditional Video Prediction using Deep Networks in Atari Games. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems, NIPS 2015, Montreal, Quebec, Canada*, 2863–2871.

Osband, I.; Aslanides, J.; and Cassirer, A. 2018. Randomized Prior Functions for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS 2018, Montréal, Canada*, 8626–8638.

Ostrovski, G.; Bellemare, M. G.; van den Oord, A.; and Munos, R. 2017. Count-Based Exploration with Neural Density Models. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia*, volume 70, 2721–2730.

Oudeyer, P.; and Kaplan, F. 2007. What is intrinsic motivation? A typology of computational approaches. *Frontiers Neurorobotics* 1: 6.

Oudeyer, P.; Kaplan, F.; and Hafner, V. V. 2007. Intrinsic Motivation Systems for Autonomous Mental Development. *IEEE Trans. Evol. Comput.* 11(2): 265–286.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia*, volume 70, 2778–2787.

Pathak, D.; Gandhi, D.; and Gupta, A. 2019. Self-Supervised Exploration via Disagreement. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA*, volume 97, 5062–5071.

Savinov, N.; Raichuk, A.; Vincent, D.; Marinier, R.; Pollefeys, M.; Lillicrap, T. P.; and Gelly, S. 2019. Episodic Curiosity through Reachability. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA*.

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

Singh, S. P. 1992. Transfer of Learning by Composing Solutions of Elemental Sequential Tasks. *Machine Learning* 8: 323–339.

Singh, S. P.; Barto, A. G.; and Chentanez, N. 2004. Intrinsically Motivated Reinforcement Learning. In *Advances in Neural Information Processing Systems, NIPS 2004, Vancouver, British Columbia, Canada*, 1281–1288.

Stadie, B. C.; Levine, S.; and Abbeel, P. 2015. Incentivizing Exploration In Reinforcement Learning With Deep Predictive Models. *CoRR* abs/1507.00814.

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

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Chen, X.; Duan, Y.; Schulman, J.; Turck, F. D.; and Abbeel, P. 2017. #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems, NIPS 2017, Long Beach, CA, USA*, 2753–2762.

Yu, X.; Lyu, Y.; and Tsang, I. W. 2020. Intrinsic Reward Driven Imitation Learning via Generative Model. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, virtual*, volume 119, 10925–10935.