DELAYED REWARDS CALIBRATION VIA REWARD EMPIRICAL SUFFICIENCY

Yixuan Liu  Hu Wang  Xiaowei Wang  Xiaoyue Sun  Liuyue Jiang
Minhui Xue
The University of Adelaide, Adelaide, Australia

ABSTRACT

Appropriate credit assignment for delay rewards is a fundamental challenge for reinforcement learning. To tackle this problem, we introduce a delay reward calibration paradigm inspired from a classification perspective. We hypothesize that well-represented state vectors share similarities with each other since they contain the same or equivalent essential information. To this end, we define an empirical sufficient distribution, where the state vectors within the distribution will lead agents to environmental reward signals in the consequent steps. Therefore, a purify-trained classifier is designed to obtain the distribution and generate the calibrated rewards. We examine the correctness of sufficient state extraction by tracking the real-time extraction and building different reward functions in environments. The results demonstrate that the classifier could generate timely and accurate calibrated rewards. Moreover, the rewards are able to make the model training process more efficient. Finally, we identify and discuss that the sufficient states extracted by our model resonate with the observations of humans.

1 INTRODUCTION

Reinforcement learning (RL) approaches have made incredible breakthroughs in various domains (Silver et al., 2016; Mnih et al., 2015; OpenAI et al., 2019; Vinyals et al., 2019), where the performance exceeds people’s expectations. The reinforcement learning theoretically models sequential decision tasks as dynamic programming processes to maximize expected accumulated rewards. Given that environmental rewards generally cannot entirely reflect the contribution of each action in a step, existing approaches commit to distributing different credits to individual decisions, known as credit assignment (Sutton & Barto, 1998). Bellman equation-based models calculate state values based on the expectation of gathered rewards from later steps, which at times assigns an unreasonable value to prior states. This problem becomes even more intractable when reward signals are extremely sparse or severely delayed.

In this paper, we formulate a purify-trained classification mechanism to extract empirical sufficient conditions of acquiring desired environmental signals as positive rewards. We refer to this extraction formulation as an Empirical Sufficient Condition Extractor (ESCE) to fairly assign delayed rewards to corresponding states. In so doing, we first propose to identify empirical sufficient states with a classification mechanism. To train the classifier with partially labeled data, we label the state vectors with predefined matched environmental signals. Then, we proposed to train the classifier with two stages, wherein a novel purified training process is conducted. In addition to existing value-based estimation, the ESCE provides concrete and objective predictions. We equip Asynchronous Advantage Actor Critic (A3C) (Mnih et al., 2016) agents with the ESCE and measure the performance on six Atari games, most of which have delayed discrete rewards. We examine the extraction correctness by formulating different reward functions, and further track the accuracy/recall changes of ESCE on the fly. The results show the agents guided by the proposed empirical efficiency achieve significant improvements in convergence, especially in the scenarios with delayed rewards. Furthermore, we constructively modify the environment to render the rewards even to be more delayed, termed as hindsight rewards settings. The results show that the calibrated rewards are able to lead agents to acquire well-learned target policies even if in the hindsight rewards scenarios. In addition to quantitative experiments, we screenshot the identified sufficient states, showing the high similar-
ity of calibrated rewards received with human’s perceptions. Our contributions can be summarized as follows:

- We introduce a model to extract empirical sufficient conditions from classification perspective, where we propose to significantly reduce the uncertainty with purified training scheme.
- We define empirical sufficient conditions and formulate a calibrated rewards in line with corresponding environmental signals to tackle the reinforcement learning reward delay issues. Such that, the agents receive rewards whenever the empirical sufficient conditions are satisfied.
- The experimental results show reward-calibrated agents are able to learn significantly better policies in the scenarios where rewards have been severely delayed than the agents without equipping the proposed calibrated reward. Moreover, we further identify and discuss the sufficient states extracted by our model resonate with the observations of human.

2 RELATED WORK

2.1 INTRINSIC MOTIVATION

Intrinsic rewards (Singh et al., 2004; Ryan & Deci, 2000) are inspired by intrinsic motivation to either encourage exploration or fulfill certain purposes. The mechanism to obtain intrinsic rewards is usually independent to it of the environmental rewards. Exploration-oriented intrinsic rewards are generally correlated to the novelty or informative acquisition of new arrival states (Pathak et al., 2017; Burda et al., 2019; Houthooft et al., 2016; Zhang et al., 2019). Because the awarding mechanism of intrinsic rewards does not depend on environments, as an exchange, the behaviour generated from a policy may not perfectly align with the final objective. In addition to exploration, intrinsic rewards can often be found in hierarchical frameworks (Kulkarni et al., 2016; Vezhnevets et al., 2017; Frans et al., 2018). Also, intrinsic rewards are used to assist agents to learn optimal or near-optimal policies in a more direct manner (Wang et al., 2020; Zheng et al., 2018; 2019). Unlike the exploration encouraged intrinsic reward, the proposed ESCE is able to generate accurate intrinsic rewards by identifying key states for better policy learning.

2.2 CREDIT ASSIGNMENT FOR DELAYED REWARDS

Most expectation estimators in reinforcement learning rely on Bellman equation, where the expectation accumulated rewards are passed through states following the basic idea of dynamic programming (Lee et al., 2019; Arjona-Medina et al., 2019; Ng et al., 1999; Marom & Rosman, 2018). To make the training more efficient, one method is to build an extra model to capture critical states for better model building (Sutton et al., 2016; Ke et al., 2018; Hung et al., 2018). Ideologically, Irgan et al. (2019) introduce binary classification into value estimation and positive-unlabeled learning (Kiryo et al., 2017) is adopted to distinguish promising and catastrophic states. Different from existing works, our work evaluates states by discriminating states as a binary classification problem without relying on Bellman equation and the idea of dynamic programming. Specially, by accurately differentiating states between “sufficient for success” and “insufficient for success”, a powerful intrinsic reward estimator is built (please refer to Sections 3.3 and 3.4 for more details).

3 DELAYED REWARD CALIBRATION

3.1 EMPIRICAL SUFFICIENT DISTRIBUTION (ESD)

The emergence of a particular state always leads to a consequence, which we refer as sufficient conditions. We consider a set of particular environmental signals as target consequences and let the stored state vectors predict these consequences to yield the sufficiency to incur these consequences. We hypothesize that closely distributed state vectors share similar information which will further incur similar results. We therefore use this hypothesis to proceed with the classification-based evaluation.
Figure 1: The overall framework of a RL agent equipped with ESCE. Agents receive environmental rewards and calibrated rewards from ESCE as its total rewards. The policy network and ESCE are two independent models. The ESCE examines each state and provides calibrated rewards whenever a state is identified as an empirical sufficient state. Within the ESCE training process, state vectors are automatically labeled and then stored in corresponding pools. The ESCE network is updated with purified training, where phase one is a binary classification with data from pools and $R_{negative}$ pools, and phase two is the proposed purified training updated with $R_{negative}$ data alone.

**Definition 3.1. Reward Empirical Sufficiency.** If there exists a continuous space such that state vectors within it incur a particular environmental consequences, we define these states as the Reward Empirical Sufficiency.

Given a stable environment, let $R_{positive}$ be the desired environmental signals, including positive rewards; conversely, $R_{negative}$ represents undesired environmental signals, including negative rewards, agent’s deaths, game endings. We further define Empirical Sufficient Distribution (ESD) as: when agents explore the environment with a specific policy, if an agent reaches a state $s^{\pi}_{suff}$ and invariably acquires $R_{positive}$, we define a state $s^{\pi}_{suf}$ as the empirical sufficient state to acquire $R_{positive}$. If the all states in a distribution are empirical sufficient states, then we consider the distribution as the Empirical Sufficient Distribution of $R_{positive}$.

### 3.2 Learning with Hybrid Reward Functions

We design Empirical Sufficient Condition Extractor (ESCE) as an independent module that can be incorporated into multiple mainstream reinforcement learning frameworks. Calibrated rewards are provided by ESCE when a state meets the empirical sufficient condition. We denote $\pi(s_t; \theta)$ as the learned policy, where $s_t$ is the observed state at time $t$ and $\theta$ is the set of parameters of the policy network; $r^c_t$ is the calibrated reward generated by ESCE at time step $t$ and $r^e_t$ is the environmental reward from the environment at time step $t$. The total reward function is synthesized with calibrated signals and environmental signals, $r_t = \alpha r^c_t + \beta r^e_t$, where $\alpha$ and $\beta$ are the weight coefficients of corresponding rewards. Our baseline, optimized by environmental rewards, has coefficients $\alpha = 0$ and $\beta = 1$. The policy network is optimized to maximize the expected accumulated rewards:

$$\pi^*(s; \theta) = \max_{\theta} \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \right] | s_t = s |.$$  \hspace{1cm} (1)

The overview of our framework is illustrated in Figure 1. Whenever an empirical sufficient state is identified by ESCE, a positive reward is offered. The calibrated reward is available until receiving an predefined positive environmental signal. Meanwhile, these states are labeled and stored in the corresponding pools classifier updating in the future. The training of the ESCE and policy network proceeds alternately.
3.3 **THE LABELING OF ESCE**

We implement ESCE training with binary classification and take individual RGB pixels of states as input. For the labeling process, we self-supervised this process to treat predefined environmental signals as classification labels.

First, we cut a long sequence of episode into several rounds along with predefined environmental signals. Later on, these environmental signals are adopted as the labels for the states within a round. Specifically, a newly received environmental signal is treated as the label for states starting from the last environmental signal until this one. The labeling procedure is shown in the Figure 1. We generalize predefined desired environmental signals as $R_{\text{positive}}$, and all undesired environmental signals as $R_{\text{negative}}$. Accordingly, $R_{\text{positive}}$ and $R_{\text{negative}}$ pools are created for labeled images collection.

As we adopt nearby environmental signals as labels, a potential problem to be solved is label ambiguity. For some states whose consequences are not determined, their labels may be different in different episodes, each beginning state for instance. As a result, the state vectors of two different labels might be densely mixed, and there might not be a clear boundary between them, which has significant difference towards normal supervised-learning scenarios. In order to resolve the label ambiguity issue, regarding Definition 3.1, ESD only contains states that lead to one particular environmental signal. It indicates that ESD should only include states with unique labels. Therefore, we formulate ESCE training as a purified training process to exclude state space with ambiguous labels.

3.4 **PURIFIED TRAINING PROCESS**

We formulate a two-phase training process to extract purified empirical sufficient distribution. In phase one, we expect ESCE to correctly identify all samples as much as possible by assigning dominant labels to the states since ambiguous states may confuse the classifier. In phase two, to exclude insufficient states, an purified training mechanism is adopted to optimize the decision boundary. We measure performance improvement of ESCE with Precision and Recall on $R_{\text{positive}}$. The calculation of Precision and Recall can be presented:

\[
\text{Precision}_{\text{pos}} = \frac{N_{\text{suff}}}{N_{\text{ident}}},
\]

\[
\text{Recall}_{\text{pos}} = \frac{N_{\text{suff}}}{N_{\text{pos}}},
\]

where the Recall indicates the ratio of identification coverage on those samples associated with $R_{\text{positive}}$, and the Precision shows how accurate the identification is. We set the number of rounds including identified states as $N_{\text{ident}}$, and let $N_{\text{sufficient}}$ denote the number of rounds that contains identified empirical sufficient states leading to $R_{\text{positive}}$. Additionally, we set the total number of samples leading to $R_{\text{positive}}$ acquired as $N_{\text{pos}}$.

The phase one is carried out by a binary classification training. The training data comes from both $R_{\text{positive}}$ and $R_{\text{negative}}$ pools, where all $R_{\text{positive}}$ samples ($s_{\text{pos}}$) are labeled with desired signals ($r_{\text{pos}}$), and $R_{\text{negative}}$ samples ($s_{\text{neg}}$) have undesired signal labels ($r_{\text{neg}}$). Both types of samples are adopted to prudently maximize the Recall of ESCE (Figure 2(a)). Let the function learner $f$ generate estimations of future rewards $\hat{r}$ with parameters $\psi$ of ESCE, where the calibrated rewards $\hat{r}_{\text{pos}}$ and $\hat{r}_{\text{neg}}$ are generated from $s_{\text{pos}}$ and $s_{\text{neg}}$, respectively:

\[
\hat{r}_{\text{pos}} = f(s_{\text{pos}}; \psi),
\]

\[
\hat{r}_{\text{neg}} = f(s_{\text{neg}}; \psi).
\]

Within the optimization to correctly classify each sample, binary cross-entropy objective is adopted. The loss of phase one measures the discrepancy between the estimated rewards and their ground truth, which are defined as follows:
Figure 2: (a) A diagram of the decision boundary improvement after phase-one of the proposed purified training: the classifier tries to make most of the samples correctly distributed on both sides of the boundary. However, there are still a few misclassified samples falsely allocated due to label ambiguity. (b) A diagram of decision boundary after phase-two of the proposed purified training: all state vectors labeled with $R_{\text{negative}}$ can be correctly classified and it matches the definition of ESD. (c) A diagram of sensitive sampling: the ESD may be changed with the updates of policy. The state vectors inside the grey region were the insufficient states, and turn into empirical sufficient states later on.

\[ L_1 = -r_{\text{pos}} \cdot \log (p(\hat{r}_{\text{pos}})) + (1 - r_{\text{neg}}) \cdot \log (1 - p(\hat{r}_{\text{neg}})) \], \quad (6) \]

In phase two, we try to maximize Precision only with $R_{\text{negative}}$ samples. The classification boundary is updated to acquire a distribution of purified $R_{\text{positive}}$ samples by excluding all insufficient samples (please refer to Figure 2(b)). The objective function adopted in phase-two is defined as follows:

\[ L_2 = -(1 - r_{\text{neg}}) \cdot \log (1 - p(\hat{r}_{\text{neg}})) \], \quad (7) \]

After the termination of phase-two, the states recognized as $R_{\text{positive}}$ should only include states with label $R_{\text{positive}}$. In other words, all states identified as $r_{\text{pos}}$ would incur $R_{\text{positive}}$, in line with the definition of empirical sufficient distributions and thus the Empirical Sufficient Distribution (ESD) is formed. We show the ESCE architecture in Algorithm 1, which can be found in the Appendix.

### 3.5 Sensitive Sampling

To efficiently update the ESCE network, we adopt a sensitive sampling strategy for purified training. ESD may change with the updates of policy network. For instance, if an policy network is significantly improved, it may stably acquire rewards which were previously unattainable. As a result, the space containing ESD might be expanded. An efficient way to update the parameter $\theta_E$ of ESCE is to pay more attention on those “hard” examples (please refer to Figure 2(c)). Accordingly, we build two extra state pools for data collection. One pool is for miss-identified samples and the other pool is built for false-identified samples. These two state pools force ESCE to focus on those “hard” examples for faster convergence and better performance. Empirically, 75% of training data are imported from two sensitive pools in our experiments.

### 4 Experimental Results

We primarily measure the effectiveness of the proposed ESCE module by adopting different reward functions with A3C-LSTM agents as the baseline on six Atari games. The original A3C-LSTM agent is optimized with environmental rewards only, which is $r_t = 0 \cdot r_c^e + 1 \cdot r_e^t$. The detailed experimental settings are laid out in Appendices A.1 and A.2. Besides, we examine the performance of the agents on new experimental settings, termed as hindsight rewards settings, where the environmental rewards is provided with more delayed time. We also examine how purified training process affects the recognition on $R_{\text{positive}}$ samples and the effectiveness of calibrated rewards with the following two questions:

- **Q1:** Does the purified training process help to improve the performance of agents?
- **Q2:** How do calibrated rewards affect the convergence of RL training?
Table 1: Comparison with baselines on hindsight rewards settings: only the calibrated rewards are provided as \( r_t^c = 1 \cdot r_t^c + 0 \cdot r_t^e \) and rewards are offered after a \( R_{\text{negative}} \) signal or at the end of episodes. The real time \( \text{Precision} \) and \( \text{Recall} \) of \( R_{\text{positive}} \) are presented in the right side of the figures.

|                | FishingDerby-v0 | Breakout-v0 | Pong-v0 |
|----------------|-----------------|-------------|--------|
| Max Score      | 5.0             | 782.0       | 3.0    |

We attempt to answer Q1 in Sections 4.1 and 4.2 and answer Q2 in Sections 4.3 and 4.4

4.1 PRECISION AND RECALL OF EMPIRICAL SUFFICIENT STATE TRAINING

To answer Q1, we first examine the training process of empirical sufficient extraction from a statistical perspective. The \( \text{Recall} \) and \( \text{Precision} \) of positive samples are a pair of trade-off. Since the \( R_{\text{positive}} \) and \( R_{\text{negative}} \) samples are densely mixed, the purified training process may excludes some \( R_{\text{positive}} \) samples when eliminates negative samples. Empirically, we fine-tuned the hyper-parameter \( \sigma \) from 0.81 to 1, which in turn keeps both of the \( \text{Recall} \) and the \( \text{Precision} \) values at a high level. The changes of these two indices are recorded in the right sides of Tables 1 and 2. We also identify that \( \text{Precision} \) is positive correlated with the improvement of the policy. This is caused by the random initialization of the policy networks, the erratic performance of the agent makes the reward prediction inaccurate. In most games, both \( \text{Recall} \) and \( \text{Precision} \) could reach high values after the convergence of the policy networks.

For Breakout-v0 shown in Table 2, the \( \text{Recall} \) is significantly reduced after the agent gets more than 40 marks on average as the bricks hit by the pellet could be vastly different in every episode. Thus, there are larger variances of Breakout-v0 when compared to other games.

4.2 THE IDENTIFICATION OF EMPIRICAL EFFICIENT STATES

To further verify the correctness of ESCE, we screenshot the extracted states on three popular games. We empirically find that the identified states have high correlations with rewards acquisition and they are visually similar to the perceptions of human. With the evolving of the policy performance, ESD keeps improving as well. In the initial episode, the empirical sufficient checkpoints are recognized a few steps away from the states where the actual rewards are given; with the policy becoming stronger and stabler, more and earlier states can be identified by ESCE. We thus infer that a well-trained ESCE model is capable of making accurate reward predictions through the understanding of the policy and environment with the training goes on. We also visualize the screenshots of extracted states in Figure 3.

4.3 DELAYED REWARD CALIBRATION IN HINDSIGHT REWARDS SETTING

The latency of rewards in realistic environments is highly unpredictable. In the reward delay environments, experimental rewards are unable to reflect the performance of policies. Therefore, we further modify the environments to force it to offer even more delayed rewards, termed as hindsight rewards. In this setting, rewards are only provided if an episode ends or after a predefined negative environmental signal. We compare the performance of agents in three games trained with environmental rewards only or combined with the proposed calibrated rewards in the hindsight rewards settings. The results are shown in Table 1. In this modified scenario, agents guided by environmental rewards only can hardly make any progress, whereas the ones updated with calibrated rewards are able to learn distinctive policies.
Figure 3: (a) The calibrated rewards and environmental rewards are presented in blue and yellow lines; the value estimation (Critic) trained with calibrated rewards and environmental rewards are denoted by green and red lines, respectively. The blue line displays that ESCE identifies an empirical sufficient state when the agent hit the pellet with the edge of the bat (right bat). It significantly increases transverse velocity for the ball which lead the agent to win the game. However, the value of baseline (red line) increases continuously until receiving the reward returned by the environment, which is far away from the decisive state. Existing approaches could not make such precise prediction (blue) on critical states. (b) In FishingDerby-v0, most states are identified as the empirical sufficient states when the hook is close to fishes or a fish is already hooked. (c) For Breakout-v0, most identified states occur when the pellet is close to the bat or the bat is on the pellet’s potential trajectory. (d) For Pong-v0, the recognized states show that the opponent is about to miss, or agents hit the pellet with the edge of the bat to give the pellet a quick vertical velocity to win the game.

Table 2: Model convergence comparison of different reward function. The agents in row-one is equipped $r_t = 0.3 \cdot r_c^t + 1 \cdot r_e^t$ and the agent in row-two is equipped with $r_t = 1 \cdot r_c^t + 0 \cdot r_e^t$. The real time Precision and Recall of $R_{positive}$ are presented in the right side of each sub-figure.

|        | FishingDerby | Breakout | Pong | Boxing | Asterix | Bowling |
|--------|--------------|----------|------|--------|---------|---------|
| Agents with Environmental Rewards | ![Graph](image1) | ![Graph](image2) | ![Graph](image3) | ![Graph](image4) | ![Graph](image5) | ![Graph](image6) |
| Agents with Customized Rewards | ![Graph](image7) | ![Graph](image8) | ![Graph](image9) | ![Graph](image10) | ![Graph](image11) | ![Graph](image12) |
| Precision | ![Graph](image13) | ![Graph](image14) | ![Graph](image15) | ![Graph](image16) | ![Graph](image17) | ![Graph](image18) |
| Recall | ![Graph](image19) | ![Graph](image20) | ![Graph](image21) | ![Graph](image22) | ![Graph](image23) | ![Graph](image24) |

The reason behind this phenomenon is that the ESCE model helps to calibrate the delayed reward by identifying those states that meet the empirical sufficient condition, which in turn speeds up the convergence of the policy networks. Given that the state value is gained from the acquired rewards, rewards received from a delayed state would thus mislead the value estimation. Figure 3(a) illustrates the difference of agent value estimation with environmental rewards and calibrated rewards. As seen from Figure 3(a), the value function (Critic) trained with environmental rewards and calibrated rewards are encoded in red and green, respectively. The calibrated reward (blue) is offered beforehand compared with the environment reward (yellow), since the ESCE identifies an empirical sufficient state at the moment when the agent hits the pellet with the edge of the bat.
4.4 Semi-Calibrated and Fully-Calibrated Rewards Settings

In addition to the hindsight rewards settings, we also conduct experiments on two reward settings to explore how the calibrated rewards affect the training process of RL agent. The first is the semi-calibrated rewards setting, where the calibrated reward coefficient is set to 0.3 and the environmental reward coefficient is 1, written as

\[ r_t = 0.3 \cdot r^{c}_t + 1 \cdot r^{e}_t. \]

The second is the fully-calibrated rewards setting, where the reward function is

\[ r_t = 1 \cdot r^{c}_t + 0 \cdot r^{e}_t. \]

The results are shown in the Table 2 as well.

For the semi-calibrated rewards setting, the results show that a small number of calibrated rewards is able to accelerate the training process of agent. It is also can be observed that the higher \textit{Recall} and \textit{Precision} values are essential to ensure the effectiveness of calibrated rewards. In Breakout-v0 and Boxing-v0, the \textit{Recall} can barely reach a high value, and we believe this is due to the large variance in the states of these games (please refer to Section 4.2).

For the fully-calibrated rewards setting, only calibrated rewards are provided. This ablation study examines the rationality of the time of awarding. The results show that agents trained with calibrated rewards can beat our baseline model in multiple different games. To acquire rewards in FishingDerby-v0, agents need to move a hook to catch fishes. Then, reel back the line before a shark eats the fish. It is common that the shark steals the fish if the agents have not learned to pull the hook up quickly. In the first column of Table 2, the calibrated rewards significantly boost the convergence speed of the model. Thus, a spike shows. This may be because when the agent learns a stable policy to pull the hook up, the states of a simple action to hook the fish becomes the empirical sufficient state and the calibrated rewards can be provided even earlier. This training process also shares similarity of the learning curve of humans, by breaking down complex missions into easy sub-tasks.

5 Conclusion

In this paper, we formulate an approach to calibrating delayed rewards from a classification perspective. Due to the purified training, the proposed ESCE model is capable of accurately extracting the critical states. Accordingly, the agents trained with calibrated rewards could be assigned with reward without delay. In addition, the results show that agents trained with the proposed calibrated rewards could learn distinctive policies in environments with extremely delayed rewards. Furthermore, we further identify and discuss the sufficient states extracted by our model resonate with the observations of human.

References

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: return decomposition for delayed rewards. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), \textit{Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada}, pp. 13544–13555, 2019.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. \textit{arXiv preprint arXiv:1606.01540}, 2016.

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In \textit{7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019}. OpenReview.net, 2019.

Kevin Frans, Jonathan Ho, Xi Chen, Pieter Abbeel, and John Schulman. Meta learning shared hierarchies. In \textit{6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings}. OpenReview.net, 2018.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: variational information maximizing exploration. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), \textit{Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, November 8-14, 2020, Virtual Event, USA}, pp. 9645–9656, 2020.
Chia-Chun Hung, Timothy P. Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value. *CoRR*, abs/1810.06721, 2018.

Alexander Irpan, Kanishka Rao, Konstantinos Bousmalis, Chris Harris, Julian Ibarz, and Sergey Levine. Off-policy evaluation via off-policy classification. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 5438–5449, 2019.

Nan Rosemary Ke, Anirudh Goyal, Olexa Bilaniuk, Jonathan Binas, Michael C. Mozer, Chris Pal, and Yoshua Bengio. Sparse attentive backtracking: Temporal credit assignment through reminding. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 7651–7662, 2018.

Ryuichi Kiryo, Gang Niu, Marthinus Christoffel du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 1675–1685, 2017.

Tejas D. Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3675–3683, 2016.

Su Young Lee, Sung-Ik Choi, and Sae-Young Chung. Sample-efficient deep reinforcement learning via episodic backward update. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 2110–2119, 2019.

Ofir Marom and Benjamin Rosman. Belief reward shaping in reinforcement learning. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3762–3769. AAAI Press, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1928–1937. JMLR.org, 2016.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Ivan Bratko and Saso Dzeroski (eds.), *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pp. 278–287. Morgan Kaufmann, 1999.
OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik’s cube with a robot hand. CoRR, abs/1910.07113, 2019. URL http://arxiv.org/abs/1910.07113.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, pp. 2778–2787. PMLR, 2017.

Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. Contemporary educational psychology, 25(1):54–67, 2000.

Richard Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavys Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. Nat., 529(7587):484–489, 2016. doi: 10.1038/nature16961.

Satinder P. Singh, Andrew G. Barto, and Nuttapon Chentanez. Intrinsically motivated reinforcement learning. In Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada], pp. 1281–1288, 2004.

Richard S. Sutton and Andrew G. Barto. Reinforcement learning - an introduction. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6.

Richard S. Sutton, Ashique Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. J. Mach. Learn. Res., 17:73:1–73:29, 2016.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, pp. 3540–3549. PMLR, 2017.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yuri Sulsky, James Molloy, Tom L. Paine, Çağlar Gülçehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft II using multi-agent reinforcement learning. Nat., 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z.

Hu Wang, Qi Wu, and Chunhua Shen. Soft expert reward learning for vision-and-language navigation. CoRR, abs/2007.10835, 2020.

Jingwei Zhang, Niklas Wetzel, Nicolai Dorka, Joschka Boedecker, and Wolfram Burgard. Scheduled intrinsic drive: A hierarchical take on intrinsically motivated exploration. CoRR, abs/1903.07400, 2019.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, pp. 4649–4659, 2018.
Zeyu Zheng, Junhyuk Oh, Matteo Hessel, Zhongwen Xu, Manuel Kroiss, Hado van Hasselt, David Silver, and Satinder Singh. What can learned intrinsic rewards capture? *CoRR*, abs/1912.05500, 2019.
A APPENDICES

A.1 SPARSE AND DELAY REWARD SETTING

We evaluate our model and the comparing models on 6 Atari 2600 games of OpenAI Gym [Brockman et al., 2016]. Positive rewards are defined as \( R_{\text{positive}} \), while the negative rewards, deaths, and game endings are denoted as \( R_{\text{negative}} \). Reinforcement Learning algorithms are inevitably facing the low sample efficiency issue, while the delayed rewards worsen the problem. Especially, in those games that rewards are offered lately, for instance, in the game Bowling-v0, the environmental rewards are determined when the character throws the ball, the proposed calibrated reward is able to mitigate the problem significantly. The longer gaps in time between empirical sufficient state and reward are, the more challenge the agent has to learn the “correct” policies.

A.2 A3C AND ESCE ARCHITECTURE

We adopt an A3C-LSTM framework as our backbone architecture. The original RGB image with size 210×160 are converted to 80×80 gray-scale frames. Four continuous frames are stacked as the input. In A3C architecture, four convolution layers and max-pooling layers are adopted. An LSTM layer with 512 units is followed with two heads — a policy head and a value function head.

Considering the policy is evolving on the fly, the empirical sufficient conditions should be updated accordingly. To ensure the extracted empirical sufficient distribution is up to date with the policy, we train the ESCE model with data sampled from the latest episodes. Subject to this reason, the maximum capacity of the datasets for the training of empirical sufficient state classifier should be flexibly decided. On the other hand, model-free reinforcement learning agents request large amount of samples for convergence. Thus, it is necessary to expand datasets to cover more cases. In our experiments, the capacity of both \( R_{\text{positive}} \) and \( R_{\text{negative}} \) pools is set from 20,000 frames to 80,000 frames, sampled by 24 workers.

Algorithm 1: Empirical Sufficient Conditions Extractor (ESCE)

Initialize Extractor network and policy \( \theta \);
Initialize \( R_{\text{positive}} \) pool and \( R_{\text{negative}} \) pool;
repeat
| Initialize temporary storage;
| while any state pool is not full do
| | if state \( S_t \) is identified as \( R_{\text{positive}} \) and no calibrated reward have been given to the agent in this round then
| | | Assign calibrated reward to the agent;
| | end
| Push states to a temporary storage;
| Update the policy network with the parameters \( \theta \);
| if environmental signal is not null then
| | Push temporary storage to pool based on positive/negative environmental signals;
| | Clear temporary storage;
| | Reset calibrated awarding status (Start a new round);
| end
| Update Extractor with states from both pools;
| while Recall of \( R_{\text{negative}} < \sigma \) do
| | Update ESCE parameters with data from \( R_{\text{negative}} \) pool;
| end
| Clear all pools;
until Converged;