# Improving On-policy Learning with Statistical Reward Accumulation

**Yubin Deng**[1] , **Ke Yu**[1] , **Dahua Lin**[1] , **Xiaoou Tang**[1] and **Chen Change Loy**[2]
[1]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[2]SenseTime-NTU Joint AI Research Centre, Nanyang Technological University
{dy015, yk017, dhlin, xtang}@ie.cuhk.edu.hk, ccloy@ntu.edu.sg

## Abstract

Deep reinforcement learning has obtained significant breakthroughs in recent years. Most methods in deep-RL achieve good results via the maximization of the reward signal provided by the environment, typically in the form of discounted cumulative returns. Such reward signals represent the immediate feedback of a particular action performed by an agent. However, tasks with sparse reward signals are still challenging to on-policy methods. In this paper, we introduce an effective characterization of past reward statistics (which can be seen as long-term feedback signals) to supplement this immediate reward feedback. In particular, value functions are learned with multi-critics supervision, enabling complex value functions to be more easily approximated in on-policy learning, even when the reward signals are sparse. We also introduce a novel exploration mechanism called "hot-wiring" that can give a boost to seemingly trapped agents. We demonstrate the effectiveness of our advantage actor multi-critic (A2MC) method across the discrete domains in Atari games as well as continuous domains in the MuJoCo environments[1]. A video demo is provided at `https://youtu.be/zBmpf3Yz8tc`.

## 1 Introduction

Advances in deep learning have mobilized the research community to adopt deep reinforcement learning (RL) agents for challenging control problems, typically in complex environments with raw sensory state-spaces. Breakthroughs by Mnih *et al.* [2015] show RL-agents can reach above-human performance in Atari 2600 games, and AlphaGo Zero Silver *et al.* [2017] becomes the world champions on the game of *Go*. Still, training RL agents is non-trivial. Off-policy methods typically require days of training to obtain competitive performance, while on-policy methods could be trapped in local minima. Recent techniques featuring on-policy learning Mnih *et al.* [2016]; Schulman *et al.* [2017]; Wu *et al.* [2017] have shown promising results in stabilizing the learning processes, enabling an agent to solve a variety of tasks in much less time. In particular, the state-of-the-art on-policy ACKTR agent by Wu *et al.* [2017] shows improved sample efficiency with the help of Kronecker-factored (K-Fac) approximate curvature for natural gradient updates, resulting in stable and effective model updates towards a more promising direction.

However, tasks with sparse rewards remain challenging to on-policy methods. An agent could take massive amount of exploration before reaching non-zero rewards; and as the agent learns on-policy, the sparseness of reward feedback (receiving all-zero rewards from most actions performed by the agent) could be malicious and render an agent to falsely predict all states in an environment towards a value of zero. As there does not exist a universal criterion for measuring "task sparseness", we show an ad-hoc metric in Figure 1 in an attempt to provide intuition. For example, we observe that the ACKTR agent is unable to receive sufficient non-zero immediate rewards that can provide instructive agent updates in Atari games "Freeway" and "Enduro", resulting in failures when solving these two games. Moreover, if ACKTR gets drawn to and trapped in unfavorable states (as in games like Boxing and WizardOfWor), it could again take long hours of exploration before the agent can get out of the local minima. Such evidence shows that on-policy agent could indeed suffer from the insufficiencies of guidance provided by the exclusive immediate reward signals from the environment.

In this paper, we introduce an effective auxiliary reward signal in tasks with sparse rewards to remedy the deficiencies of learning purely from standard immediate reward feedbacks. As on-policy agents may take many explorations before reaching non-zero immediate rewards, we argue that we can leverage past reward statistics to provide more instructive feedbacks to agents in the same environment. To this end, we propose to characterize the past reward statistics in order to gauge the "long-term" performance of an agent (detailed in Section 4). After performing an action, an agent will receive a long-term reward signal describing its past performance upon this step, as well as the conventional immediate reward from the environment. To effectively characterize the long-term performance of the agent, we take into considerations both the crude amount of rewards and the volatility of rewards received in the past, where highly volatile distri-

[1]Supplementary material: `http://bit.ly/supp-id4`

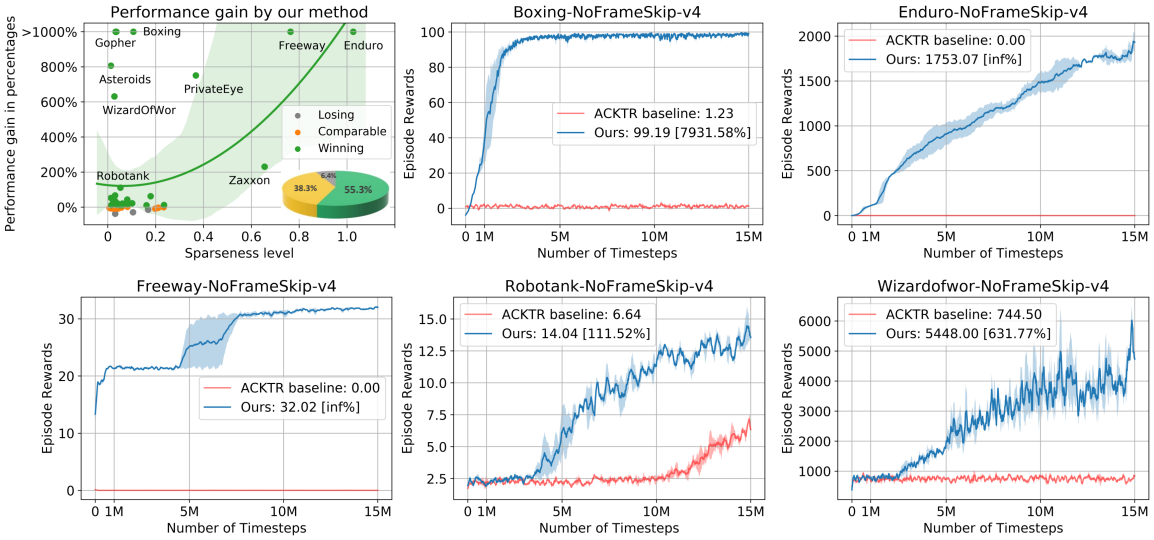


Figure 1: Performance of A2MC on Atari games trained with 15 million timesteps. Our method has a winning rate of $55.3\%$ among all the Atari games tested, as compared to the ACKTR. Our A2MC learns quickly in some of the hardest games for on-policy methods, such as "Boxing", "Enduro", "Freeway", "Robotank" and "WizardOfWor". The sparseness of a game is defined as the sparseness of average rewards $\mathbf{x}$ obtained by ACKTR within the first $n = 10^6$ timesteps by $\varphi(\mathbf{x}) = \left(\sqrt{n} - \frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_2}\right)/(\sqrt{n}-1)$. A higher value of sparseness indicates sparser rewards. A relative performance margin (in terms of final reward) larger than $10\%$ is deemed as winning / losing. The shaded region denotes the standard deviation over 2 random seeds.

butions of long-term rewards are explicitly penalized. This enables complex value functions to be more easily approximated in multi-critics supervision. We find in practice that by explicitly penalizing highly volatile long-term rewards while maximizing the expectation of short-term rewards, the learning process and the overall performance are improved regarding both sample efficiency and final rewards. We further propose a "hot-wiring" exploration mechanism that can boost seemingly trapped agent in the earlier stage of learning. By leveraging the characterization of long/short-term reward statistics, our proposed advantage actor multi-critic model (A2MC) shows significantly improved performance on the Atari 2600 games and the MuJoCo tasks as compared to the state-of-the-art on-policy methods.

## 2 Related Work

**Reward shaping and pseudo-rewards**: To tackle the challenge in tasks with rarely observed rewards, pseudo-rewards maximization is adopted in earlier works Konidaris and Barto [2009]; Silver and Ciosek [2012]. Auxiliary vision tasks (e.g., learning pixel changes or network features) are adopted in the off-policy *UNREAL* agent Jaderberg *et al.* [2016] in order to facilitate learning better feature representations, particularly for sparse reward environments. Another direction of effort involves directly engineering a better reward function or shaping the reward signal. Andrychowicz *et al.* [2017] enhances off-policy learning by re-producing informative reward in hindsight for sequences of actions that do not lead to success previously. The HRA approach Van Seijen *et al.* [2017] exploits domain knowledge to define a set of environment-specific rewards based on reward categories. And the winning approach that learns playing "Doom" Lample and Chaplot [2017] shows promising success in the FPS game that carefully crafting the task rewards would indeed be beneficial. In contrast to heuristically defining vision-related auxiliary tasks, our proposed A2MC agent learns from the characterization of past reward statistics obtainable from any environment; and different from the hybrid architecture pertaining to Ms. Pacman only and the reward shaping settings tailored specifically to "Doom", our proposed reward characterization mechanism is generic and our A2MC generalizes well to a variety of tasks without the need to engineer a decomposition of problem-specific environment rewards. Moreover, the capability of the proposed method to further boost reward shaping is evidenced in our case study on playing Doom (see Supplementary Section F).

**Multi-agents:** The *multi-agent* approaches Lanctot *et al.* [2017] present another promising direction for learning. They propose to train multiple agents in parallel when solving a task, and to combine multiple action-value functions with a centralized action-value function. The multi-critics supervision in our proposed A2MC model can be seen as a form of joint-task or multi-task learning Teh *et al.* [2017] for both long-term and short-term rewards.

**On-policy v.s. Off-policy:** Our empirical results based on learning the characterization of long/short-term reward statistics also echo the effectiveness of a recently proposed off-policy reinforcement learning framework Bellemare *et al.* [2017] that features a distributional variant of Q-learning, wherein the value functions are learned to match the distribution of standard immediate returns. Also, Wang *et al.* [2016] shows that applying experience replay to on-policy methods can further enhance learning stability. Schulman *et al.* [2016] proposes a variant of advantage function using *eligibility*

*traces*[2] that provides both low-variance and low-bias gradient estimates. Thomas et al. Thomas *et al.* [2015] propose an off-policy method for computing a lower confidence bound on the expected return of a policy for the policy evaluation problem, while our method is targeted explicitly for *policy learning* (*i.e.,* the policy control problem). These works are orthogonal to our approach can potentially be combined with the proposed characterization of past reward statistics to further enhance learning performance. Using risk-sensitive objectives Chow *et al.* [2015]; Chow and Ghavamzadeh [2014]; Tamar *et al.* [2014] has shown success in terms of robustness to modeling errors on customized setups. And while there are indeed numerous works that try to address also the sparse reward setting, we set out to base our arguments within the *on-policy* RL domain and aim for a self-contained paper on improving on-policy learning in general, wherein our method reveals effectiveness for sparse reward games. Our extensive experiments (see also Supplementary Section E and F) show promising results of our approach in both on- and off-policy frameworks, and we choose to focus on "on-policy" methods (i.e., those that do not involve off-policy trajectories or experience replay) as in Wu *et al.* [2017] in the main text in order to systematically evaluate the potential of our proposed reward mechanism within the scope of this work.

## 3 Preliminary

Consider the standard reinforcement learning setting where an agent interacts with an environment over a number of discrete time step. At each time step $t$, the agent receives an environment state $s_t$, then executes an action $a_t$ based on policy $\pi_t$. The environment produces reward $r_t$ and next state $s_{t+1}$, according to which the agent gets feedback of its immediate action and will decide its next action $a_{t+1}$. The process $< \mathbf{S}, \mathbf{A}, \mathbf{R}, \mathbf{S} >$, typically considered as a Markov Decision Process, continues until a terminal state $s_T$ upon which the environment resets itself and produces a new episode. Under conventional settings, the return is calculated as the discounted summation of rewards $r_t$ accumulated from time step $t$ onwards $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$. The goal of the agent is to maximize the expected return from each state $s_t$ while following policy $\pi$. Each policy $\pi$ has a corresponding action-value function defined as $Q^\pi(s,a) = \mathbb{E}[R_t | s_t = s, a_t = a; \pi]$. Similarly, each state $s \in S$ under policy $\pi$ has a value function defined as: $V^\pi(s) = \mathbb{E}[R_t | s_t = s]$. In **value-based approaches** (e.g., Q-learning Mnih *et al.* [2015]), function approximator $Q(s,a;\theta)$ can be used to approximate the optimal action value function $Q^*(s,a)$. This is conventionally learned by iteratively minimizing the below loss function:

$$L(\theta) = \mathbb{E}[(y_t^{target} - Q(s_t, a_t; \theta))^2], \tag{1}$$

where $y_t^{target} = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta)$ and $s_{t+1}$ is the next state following state $s_t$.

In **policy-based approaches** (e.g., policy gradient methods), the optimal policy $\pi^*(a|s)$ is approximated using the approximator $\pi(a|s;\theta)$. The policy approximator is then learned by gradient ascent on $\nabla_\theta \mathbb{E}[R_t] \approx \nabla_\theta \log \pi(a_t|s_t;\theta) R_t$. The REINFORCE method Williams [1992] further incorporates a baseline $b(s_t)$ to reduce the variance of the gradient estimator: $\nabla_\theta \mathbb{E}[R_t]_{REINFORCE} \approx \nabla_\theta \log \pi(a_t|s_t;\theta)(R_t - b(s_t))$

In **actor-critic based approaches**, the variance reduction further evolves into the advantage function $A(s_t,a_t) = Q(s_t,a_t) - V(s_t)$ in Mnih *et al.* [2016], where the action value $Q^\pi(s_t,a_t)$ is approximated by $R_t$ and $b(s_t)$ is replaced by $V^\pi(s_t)$, deriving the advantage actor-critic architecture where actor-head $\pi(\cdot|s)$ and the critic-head $V(s)$ share low-level features. The gradient update rule w.r.t. the action-head is $\nabla_\theta \log \pi(a_t|s_t;\theta)(R_t - V(s_t;\theta))$. The gradient update w.r.t. the critic-head is: $\nabla_\theta (R_t - V(s_t;\theta))^2$, where $R_t = r_t + \gamma V(s_{t+1})$.

## 4 Characterization of Past Reward Statistics

The conventional reward $r_t$ received from the environment at time step $t$ after an action $a_t$ is performed represents the immediate reward regarding this particular action. This "immediacy" could be interpreted as a *short-term* horizon of how the agent is doing, *i.e.,* evaluating the agent via judging its actions by immediate rewards. We argue that the deficiencies of learning solely from immediate rewards mainly come from this limitation that the agent is learning from one single type of exclusive short-term feedback.

As the goal of providing reward feedback to an agent is to inform the agent of its performance, we seek to find an auxiliary performance metric that can measure whether the agent is performing *consistently* well. Inspired by the formulation of *Sharpe Ratio* ($\mathbb{E}[r] \times \frac{1}{\sigma_r}$) in evaluating the long-term performance of porfolio strategies where the return $\mathbb{E}[r]$ is inversely weighted by the risk $\sigma_r$, an effective characterization of historical reward statistics should take into account at least two factors, namely 1) how high the immediate reward is and 2) how varied past rewards were, bringing the desired notion of "risk-adjusted return" as in Sharpe [1994].

### 4.1 Variability-Weighted Reward

To this end, we follow insights behind Dowd; Sharpe [1994] and define a variability-weighted characterization of past rewards. This is illustrated in Figure 2. We consider a historical sequence of $T$ rewards upon timestep $t$ (looking backward $T-1$ timesteps): $\vec{\mathbf{r}} = [r_{t-(T-1)}..., r_{t-2}, r_{t-1}, r_t]$. In order to evaluate how high and varied the reward sequence is, a few steps of pre-processing $\mathcal{G}$ is applied, denoted as $\vec{\mathcal{R}} = \mathcal{G}(\vec{\mathbf{r}})$. Specifically, we first derive the reward change at each timestep (similar to the "differential return" concept in Sharpe [1994]) with $d_n = r_n - r_{n-1}$:

$$\begin{aligned} \vec{\mathbf{d}} &= [d_{t-(T-1)}, d_{t-(T-2)}, \ldots, d_t] \\ &= [r_{t-(T-1)}, r_{t-(T-2)} - r_{t-(T-1)}, \ldots, r_t - r_{t-1}]. \end{aligned} \tag{2}$$

Then we re-order the sequence by flipping [3] with $f_n = d_{t+1-n}$:

$$\vec{\mathbf{f}} = [f_1, f_2, \ldots, f_T] = [d_t, d_{t-1}, \ldots, d_{t-(T-1)}]. \tag{3}$$

[2] See also Supplementary Section G for details regarding our method and eligibility traces.

[3] By flipping, we further encourage "recent" stable rewards and penalize the volatility of recent past rewards. A concrete example is given in the Supplementary Section A.

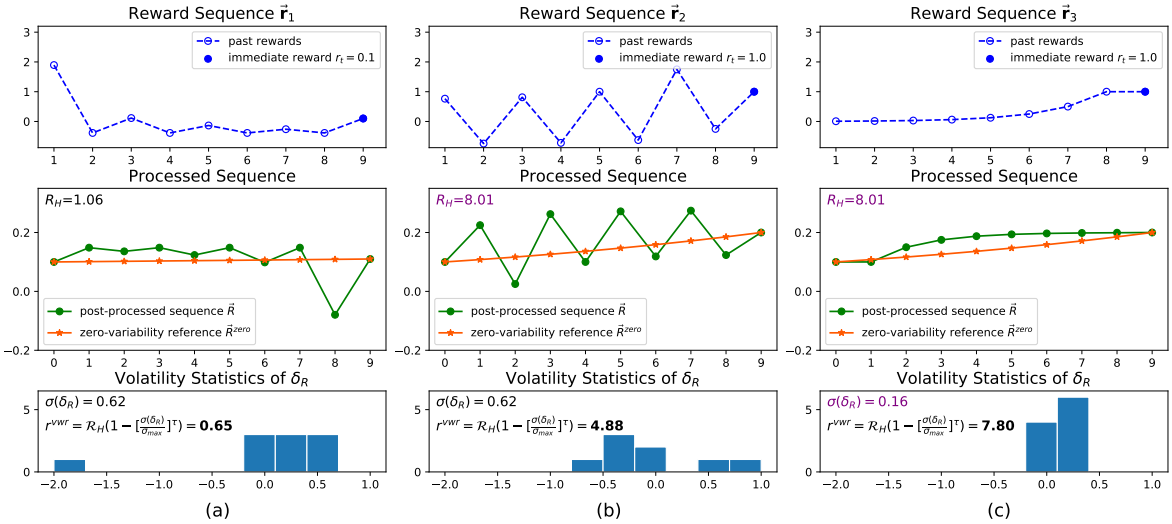


Figure 2: Illustration of the proposed variability-weighted reward (VWR). The first row shows the raw reward sequence (blue) while the second row presents the post-processed sequence $\vec{\mathcal{R}}$ (green) and the zero-variability reference $\vec{\mathcal{R}}^{zero}$ (orange), and $\mathcal{R}_H$ is calculated as a reflection of *how high the immediate reward is*. The third row shows the volatility statistics of $\delta_{\mathcal{R}}$, representing *how varied past rewards were*. We curated 3 hypothetical reward sequences – (a) highly varied sequence with low immediate reward, resulting in the lowest VWR; (b) highly varied sequence with high immediate reward, leading to a relatively high VWR; (c) stable sequence with high immediate reward, achieving the best VWR. More examples can be found in the Supplementary Section A.

Next we append $f_0 = 1$ to the head of sequence $\vec{\mathbf{f}}$ and take the normalized cumulative sum to obtain the post-processed reward sequence as $\vec{\mathcal{R}} = [\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_T] = \frac{1}{T+1}[f_0, f_0 + f_1, \ldots, \sum_{i=0}^{T} f_i]$. Under such processing, numerical instability (see Eq. 4) when all rewards in the sequence are zero can be alleviated, while the averaging term $\frac{1}{T+1}$ mitigates the effect of introducing the artificial $f_0$. Optionally, smoothing techniques such as exponential moving average may be applied as well.

The resulting $\vec{\mathcal{R}}$ is a reward sequence with $\mathcal{R}_T - \mathcal{R}_0 = \frac{1}{T+1} r_t$, and $\mathcal{R}_n - \mathcal{R}_{n-1} = \frac{1}{T+1}(r_{t+1-n} - r_{t-n})$. Therefore, the difference between $\mathcal{R}_T$ and $\mathcal{R}_0$ represents the immediate reward and the whole sequence $\vec{\mathcal{R}}$ reflects the volatility of past rewards. In Figure 2, three examples of processed sequence are presented in the second row with the corresponding raw rewards shown in the first row. We account for *how high the immediate reward is* by defining the relative percentage log total return as:

$$\mathcal{R}_H = \frac{{\mathcal{R}_T}^{1/T} - {\mathcal{R}_0}^{1/T}}{{\mathcal{R}_0}^{1/T}} \tag{4}$$

To account for *how varied past rewards were*, we first define a smooth *zero-variability reference* as: $\vec{\mathcal{R}}^{zero} = [\mathcal{R}_0^{zero}, \mathcal{R}_1^{zero}, \ldots, \mathcal{R}_T^{zero}] = \mathcal{R}_0[e^{0\times\widetilde{\mathcal{R}}}, e^{1\times\widetilde{\mathcal{R}}}, \ldots, e^{T\widetilde{\mathcal{R}}}]$ with $\widetilde{\mathcal{R}} = \frac{1}{T}\ln\frac{\mathcal{R}_T}{\mathcal{R}_0}$, representing a smooth monotonic reference sequence from $\mathcal{R}_0$ to $\mathcal{R}_T$. Then we define the reward differential $\delta_{\mathcal{R}}$ as the differential reward versus its zero-variability reference as $\delta_{\mathcal{R}}(n) = \frac{\mathcal{R}_n - \mathcal{R}_n^{zero}}{\mathcal{R}_n^{zero}}$, whose statistics are sketched in the third row of Figure 2. With maximally allowed volatility as $\sigma_{max}$, the variability weights can be defined as: $\omega = 1 - [\frac{\sigma(\delta_{\mathcal{R}})}{\sigma_{max}}]^\tau$, where $\sigma(\cdot)$ is the standard deviation and $\tau$ controls the rate to penalize highly volatile reward distribution. Finally we can derive the variability-weighted past reward indicator $r^{vwr}$ for the characterization of past reward statistics:

$$r^{vwr} = \begin{cases} \mathcal{R}_H(1 - [\frac{\sigma(\delta_{\mathcal{R}})}{\sigma_{max}}]^\tau) & \text{if } \sigma(\delta_{\mathcal{R}}) < \sigma_{max}, \mathcal{R}_T > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The formulation of Equation 5 share principled themes as in Sharpe [1994] and Dowd:

1. Dowd compares the newly obtained $\text{SR}^{new}$ with the previous $\text{SR}^{old}$ in choosing new assets; we derive $\mathcal{R}_H$ in Eq. 4 by comparing the latest reward $\mathcal{R}_T$ with $\mathcal{R}_0$ to explicitly encourage the agent to aim for reward improvements in "choosing new actions";
2. Both the Sharpe Ratio (SR) and Eq. 5 involve "variability weights" to adjust for the unit risk of return $\mathbb{E}[\mathcal{R}]$ Sharpe [1994] (i.e., $\frac{1}{\sigma_r}$ for SR and $1 - [\frac{\sigma(\delta_{\mathcal{R}})}{\sigma_{max}}]^\tau$ for $r^{vwr}$);
3. Whereas Dowd introduces the concept of "minimum required return" based on the elasticity of value at risk (VaR), we consider the maximum tolerance level $\sigma_{max}$ with elasticity controlled by $\tau$ for improved learning stability of $r^{vwr}$(see also Supplementary Section H).

Example computed values of $r^{vwr}$ for the characterization of different reward statistics are shown in Figure 2 and we show strong empirical results (in Section 6) to confirm the validity and robustness of the proposed formulation in multiple reinforcement learning domains.

## 4.2 Multi-Critic Architecture

A higher value of $r^{vwr}$ indicates better agent performance as the result of the historical sequence of actions. The same

set of optimization procedures for conventional value function (*i.e.,* via maximization of immediate reward signal $r$) update can be applied accordingly. The actual returns computed from both the "long-term" and "short-term" rewards are discounted by the same factor $\gamma$. In particular, for standard $N$-step look-ahead approaches, we have:

$$R_t^{\text{short-term}} = \sum_{n=0}^{N-1} \gamma^n r_{t+n} + \gamma^N V(s_{t+N}),$$
$$R_t^{\text{long-term}} = \sum_{n=0}^{N-1} \gamma^n r_{t+n}^{vwr} + \gamma^N V^{vwr}(s_{t+N}) \tag{6}$$

Similar to the standard state value function $V(s)$, we further define $V^{vwr}(s)$ as the value function w.r.t the variability-weighted reward $r^{vwr}$. These value functions form *multiple* critics judging a given state $s$. The gradients w.r.t. the critics then become:

$$\nabla_{critic} = \nabla_{\theta^{\text{short-term}}}[(R_t^{\text{short-term}} - V(s_t;\theta^{\text{short-term}}))^2] + \nabla_{\theta^{\text{long-term}}}[(R_t^{\text{long-term}} - V^{vwr}(s_t;\theta^{\text{long-term}}))^2] \tag{7}$$

where the standard grading clipping approach can be applied in Eq. 7 for enhanced stability. More advanced methods for estimating $R_t^{\text{short-term}}$ and $R_t^{\text{long-term}}$ above, such as the online variant of generalized advantage estimation (GAE) using eligibility traces Schulman *et al.* [2016] can be adopted in place of Eq. 6, as shown below (see also Supplementary Section G):

$$A_t^{\text{short-term}} = \sum_{n=0}^{\infty} (\gamma\lambda)^n \delta_{t+n} \text{ and } A_t^{\text{long-term}} = \sum_{n=0}^{\infty} (\gamma\lambda)^n \delta_{t+n}^{vwr}$$
$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$
$$\delta_t^{vwr} = r_t^{vwr} + \gamma V^{vwr}(s_{t+1}) - V^{vwr}(s_t) \tag{8}$$

where the generalized estimator of the advantage function $A_t^{\text{short-term}}$ and $A_t^{\text{long-term}}$ allows a trade-off of bias *v.s.* variance using the parameter $0 \leq \lambda \leq 1$, similar to the TD($\lambda$) approach for eligibility traces. We show the effectiveness of the proposed characterization of past reward statistics in multiple advantage actor-critic frameworks (i.e., ACKTR and PPO), where the two different value functions can share the same low-level feature representation, enabling a single agent to learn multiple critics parameterized by $\theta^j, j \in \{\text{short-term}, \text{long-term}\}$. The gradient to the actor branch is from both of the critics by accumulating the gradients following standard multi-task learning approaches. (See also Supplementary Section I for the full algorithm).

## 5 Hot-Wire $\epsilon$-Exploration

Being handed a game-stick, a human most likely would try out all the available buttons on it to see which particular button entails whatever actions on the game screen, hence receiving useful feedbacks. Inspired by this, we propose to hot-wire the agent to perform an identical sequence of randomly chosen actions in the N-step look-ahead during the initial stage

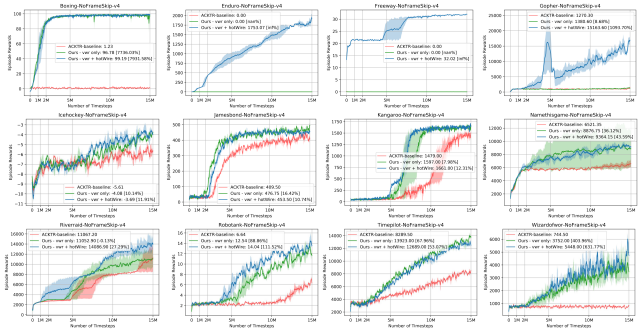

Figure 3: Performance of A2MC on Atari games. "Hot-Wiring" exploration makes the agent easier to figure out how to play challenging games like "Robotank" and "WizardOfWor", and for most games, it provides a better initial state for the agent to start off at a game and hence to obtain better final results. The number in figure legend shows the average reward among the last 100 episodes and the percentage shows the performance margin as compared to ACKTR. The shaded region denotes the standard deviation over 2 random seeds. (Best viewed with zoom-in.)

(randomly pressing down a game-stick button for a while):

$$a_{t+k} = \begin{cases} \text{a random action identical for all k} & \text{prob} = \epsilon \\ \pi(a_{t+k}|s_{t+k}) \text{ for } k = 0, ..., N-1 & \text{prob} = 1-\epsilon \end{cases} \tag{9}$$

We show that by enabling the "hot-wiring" mechanism[4], a seemingly trapped agent can be boosted to learn to quickly solve problems where rewards can only be triggered by particular action sequences, as shown in games like "Robotank" and "WizardOfWor" in Figure 3.

## 6 Experiments

We use the same network architecture and natural gradient optimization method as in the ACKTR model Wu *et al.* [2017]. We set $\sigma_{max} = 1.0$, $\tau = 2.0$ and $T = 20$ in the computation of variability-weighted reward (see Supplementary Section C for hyperparameter studies). For hot-wiring exploration, we choose $\epsilon = 0.20$ and initial stage to be first $\frac{1}{40}$ of the total training period for all experiments. Other hyperparameters such as learning rate and gradient clipping remain the same as in the ACKTR model Wu *et al.* [2017], in addition to adopting GAE Schulman *et al.* [2016] for a stronger ACKTR baseline (see Sec 4.2). We first present results of evaluating the proposed A2MC model in two standard benchmarks, the discrete Atari experiments and the continuous MuJoCo domain. Then we show ablation studies on the robustness of the hyper-parameters involved as well as evaluating the extensibility of the proposed long/short-term reward characterizations to other on-policy methods. Further extensions to off-policy domains are presented in Supplementary Section E and Supplementary Section F.

We further expand the training budget and continue learning the games until 50 million timesteps as in Wu *et al.* [2017]. As shown in Table 1, our A2MC model can solve

[4]hot-wire is triggered only when the agent is unable to receive meaningful rewards in an initial learning stage. The legend "vwr + hotWire" in Fig. 3 indicates that the mechanism is "enabled" but not "enforced".

Table 1: Comparison of average episode rewards at the end of 50 million timesteps in Atari experiments. The reward scores and the first episodes reaching human-level performance Mnih *et al.* [2015] are reported as in Wu *et al.* [2017]. A2MC is able to solve games that are challenging to ACKTR and also retain comparable performance in the rest of games.

| | | ACKTR | | A2MC | |
|---|---|---|---|---|---|
| Domain | Human | Rewards | Eps | Rewards | Eps |
| Asteroids | 47388.7 | 34171.0 | N/A | **830232.5** | **11314** |
| Beamrider | 5775.0 | 13581.4 | 3279 | 13564.3 | **3012** |
| Boxing | 12.1 | 1.5 | N/A | **99.1** | **158** |
| Breakout | 31.8 | **735.7** | 4097 | 411.4 | **3664** |
| Double Dunk | -16.4 | -0.5 | 742 | **21.3** | **544** |
| Enduro | 860.5 | 0.0 | N/A | **3492.2** | **730** |
| Freeway | 29.6 | 0.0 | N/A | **32.7** | **1058** |
| Pong | 9.3 | 20.9 | 904 | 19.4 | **804** |
| Q-bert | 13455.0 | 21500.3 | **6422** | **25229.0** | 7259 |
| Robotank | 11.9 | 16.5 | - | **25.7** | 4158 |
| Seaquest | 20182.0 | 1776.0 | N/A | **1798.6** | N/A |
| Space Invaders | 1652.0 | **19723.0** | 14696 | 11774.0 | **11064** |
| Wizard of Wor | 4756.5 | 702 | N/A | **7471.0** | **8119** |

games like "Boxing", "Freeway" and "Enduro" that are challenging for the baseline ACKTR model. For a full picture of model performance in Atari games, A2MC has a human-level performance rate of $74.5\%$ (38 out of 51 games) in the Atari benchmarks, compared to $63.6\%$ reached by ACKTR. Individual game scores for all the Atari games are reported in the Supplementary Section B.

### 6.1 ATARI 2600 Games

We follow standard evaluation protocol to evaluate A2MC in a variety of Atari game environments (starting with 30 no-op actions). We train our models for 15 million timesteps for each game environment and score each game based on the average episode rewards obtained among the last 100 episodes as in Wu *et al.* [2017]. The learning results on 12 Atari games are shown in Figure 3 where we also included an ablation experiment of A2MC without hot-wiring. We observe that on average A2MC improves upon ACKTR in terms of final performance under the same training budget. Our A2MC is able to consistently improve agent performance based on the proposed characterization of reward statistics, hence the agent is able to get out of local minima in less time (higher sample efficiency) compared to ACKTR. The complete learning results on all games are attached in the Supplementary Section B.

### 6.2 Continuous Control

For the evaluations on continuous control tasks simulated in MuJoCo environment, we first follow Wu *et al.* [2017] and tune a different set of hyper-parameters from Atari experiments. Specifically, all MuJoCo experiments are trained with a larger batch size of 2500. The results of eight MuJoCo environments trained for 1 million timesteps are shown in Figure 4. We observe that A2MC also performs well in all MuJoCo continuous control tasks. In particular, A2MC has brought significant improvement on the tasks of *HalfCheetah*, *Swimmer* and *Walker2d* (see Table 2).

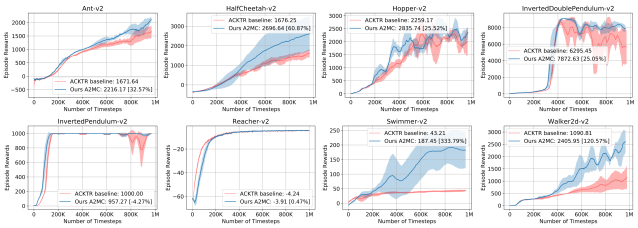


Figure 4: Performance on the MuJoCo benchmark. A2MC is also competitive on MuJoCo continuous domain when compared to ACKTR. The shaded region denotes std over 3 random seeds. (Best viewed with zoom-in.)

To test the robustness of A2MC, we perform another set of evaluations on MuJoCo tasks by keeping an identical set of hyper-parameters used in the Atari experiments. Figure 7 in Supplementary Section C shows this ablation result. We observe that even under sub-optimal hyper-parameters, our A2MC model can still learn to solve the MuJoCo control tasks in the long run. Moreover, it is less prone to overfitting when compared to ACKTR under such "stress testing". Additional hyper-parameter studies can be found in Supplementary Section C.

We also evaluate a multi-critics variant of the proximal policy optimization (PPO) model on the MuJoCo tasks with our proposed long/short-term rewards. In particular, we observe that our proposed variability-weighted reward generalizes well with the vanilla PPO, and our multi-critics PPO variant (MC-PPO) shows more favorable performance, as shown in Table 2. Specifically, MC-PPO shows the best performance on *Hopper* and *Walker-2d* among all models under the 1-million timesteps training budget. Both of our multi-critics variants (A2MC and MC-PPO) have won 6 out of the 8 MuJoCo tasks with relative performance margins (percentages in parentheses) larger than $25\%$ (see Table 2).

## 7 Conclusion

In this work, we introduce an effective auxiliary reward signal to remedy the deficiencies of learning solely from the standard environment rewards. Our proposed characterization of past reward statistics results in improved learning and higher sample efficiencies for on-policy methods, especially in challenging tasks with sparse rewards. Experiments on both discrete tasks in Atari environment and MuJoCo continuous control tasks validate the effectiveness of utilizing the proposed long/short-term reward statistics for on-policy methods using multi-critic architectures. This suggests that expanding the form of reward feedbacks in reinforcement learning is a promising research direction.

## References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NeurIPS*, 2017.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv:1707.06887*, 2017.

Table 2: Average episode rewards obtained among the last 10 episodes upon 1 million timesteps of training in MuJoCo experiments. We also extract the results from PPO+LIRPG Zheng *et al.* [2018] for completeness of comparison.

| GAMES | ACKTR | Our A2MC | PPO | PPO+LIRPG | Our MC-PPO |
|---|---|---|---|---|---|
| Ant | 1671.6 | **2216.1 (32.5%)** | 411.4 ($\pm$ 107.7) | $\sim -50$ | **618.9 (50.4%)** |
| HalfCheetah | 1676.2 | **2696.6 (60.8%)** | 1433.7 ($\pm$ 83.9) | $\sim 2000$ | **2473.4 (72.5%)** |
| Hopper | 2259.1 | **2835.7 (25.5%)** | 2055.8 ($\pm$ 274.6) | $\sim 2200$ | **3131.3 (52.3%)** |
| Inv. D-Pendulum | 6295.4 | **7872.6 (25.0%)** | 4454.1 ($\pm$ 1098.1) | N/A | **7648.7 (71.7%)** |
| Inv. Pendulum | 1000.0 | 957.2 (-4.2%) | 839.7 ($\pm$ 127.1) | N/A | 777.4 ($-7.4\%$) |
| Reacher | -4.2 | -3.9 (0.4%) | -5.47 ($\pm$ 0.3) | N/A | $-10.3$ ($-8.5\%$) |
| Swimmer | 43.2 | **187.4 (333.7%)** | 79.1 ($\pm$ 31.2) | N/A | **102.9 (30.2%)** |
| Walker2d | 1090.8 | **2405.9 (120.5%)** | 2300.8 ($\pm$ 397.6) | $\sim 2100$ | **3718.1 (61.6%)** |
| Win — Fair — Lose | N/A | 6 — 2 — 0 | N/A | N/A | 6 — 2 — 0 |


Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for cvar optimization in mdps. In *NeurIPS*, pages 3509–3517, 2014.

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. In *NeurIPS*, 2015.

Kevin Dowd. Adjusting for risk:: An improved sharpe ratio. *International review of economics & finance*.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. 2018.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv:1611.05397*, 2016.

George Konidaris and Andrew G Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *NeurIPS*, 2009.

Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning. In *AAAI*, 2017.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Julien Perolat, David Silver, Thore Graepel, et al. A unified game-theoretic approach to multiagent reinforcement learning. In *NeurIPS*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *ICLR*, 2016.

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

William F Sharpe. The sharpe ratio. *Journal of portfolio management*, 1994.

David Silver and Kamil Ciosek. Compositional planning using optimal option models. *arXiv:1206.6473*, 2012.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Aviv Tamar, Yonatan Glassner, and Shie Mannor. Policy gradients beyond expectations: Conditional value-at-risk. *arXiv:1404.3862*, 2014.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *NeurIPS*, 2017.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *AAAI*, 2015.

Harm Van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid reward architecture for reinforcement learning. In *NeurIPS*, 2017.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv:1611.01224*, 2016.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*. Springer, 1992.

Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *NeurIPS*, 2017.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In *NeurIPS*, 2018.