A Sandpile Model for Reliable Actor-Critic Reinforcement Learning

Yiming Peng, Gang Chen, Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: {yiming.peng, aaron.chen, mengjie.zhang}@ecs.vuw.ac.nz

Shaoning Pang
Department of Computing
Unitec Institute of Technology
Auckland, New Zealand
Email: ppang@unitec.ac.nz

***Abstract*—Actor-Critic algorithms have been increasingly researched for tackling challenging reinforcement learning problems. These algorithms are usually composed of two distinct learning processes, namely actor (a.k.a, policy) learning and critic (a.k.a, value function) learning. Actor learning is heavily dependent on critic learning; particularly unreliable critic learning due to its divergence can significantly affect the effectiveness of actor-critic algorithms. To address this issue, many successful algorithms have been developed recently with the aim of improving the accuracy of value function approximation. However, these algorithms introduce extra complexities to the learning process and may actually increase the difficulty for effective learning. Thus, in this research, we consider a simpler approach to improving the critic learning *reliability*. This approach requires us to seamlessly integrate an adapted Sandpile Model with the critic learning process so as to achieve desirable self-organizing property for reliable critic learning. Following this approach, we propose a new actor-critic learning algorithm. Its effectiveness and learning reliability have been further evaluated experimentally. As strongly demonstrated in the experiment results, our new algorithm can perform much better than traditional actor-critic algorithms. Meanwhile, correlation analysis further suggests that a strong correlation exists in between learning reliability and effectiveness. This finding may be important for future development of powerful reinforcement learning algorithms.**

I. Introduction

Reinforcement Learning (RL) has been playing an important role in many disciplines, including psychology, cognitive science, behavior economics and neuroscience [1]. RL is a learning paradigm in which an agent learns from its iterative interactions with the unknown environment [2]. For each interaction, an intelligent agent follows a policy to take an action at one state; meanwhile, it receives an instant reward from the environment to criticize the action. The agent aims to find an optimal policy that maximizes its cumulative rewards.

As important methods for RL, Actor-Critic algorithms (i.e., ACRL) currently are receiving a lot of research interests [2]–[4]. ACRL algorithms typically consist of two parts, namely *actor* (a.k.a, policy) and *critic* (a.k.a, value function). The actor is a mapping that recommends an action to take at any state. The critic is used to criticize the quality of the actor and approximates the cumulative rewards obtainable via the actor.

In ACRL, learning is usually conducted through two separated learning processes, one for critic learning and the other for actor learning [4]–[6]. Specifically for effective critic learning, the critic is usually represented as a parametric value function. This function is made up of two important components, i.e., *value function parameters* and *basis functions* (a.k.a, state features). They follow a linear relationship to approximate the expected cumulative rewards. Similarly, the actor learning is responsible for learning a policy which is often represented as a parametric function governed by a set of *policy parameters*. Accordingly, the gradient ascent technique is adopted for the actor learning, where the gradients of the expected cumulative rewards with respect to policy parameters are known as *policy gradients*. In practice, unbiased estimation of the policy gradients is usually determined by the learned critic.

Apparently, the overall effectiveness of ACRL largely depends on critic learning. Traditionally, aimed at improving the approximation accuracy for critic learning, many researchers introduced various mean-squared error (MSE) criteria [3], [7], [8]. For example, based on the criterion of the mean-square Bellman error (MSBE), Sutton and his colleagues proposed the Gradient Temporal Difference (GTD) algorithm in [7]. However, GTD often exhibits unreliable behavior in practice due to the divergence of the critic. To address this issue, Sutton et al. [8] proposed an important solution where a new algorithm called GTD2 was introduced by changing the learning objective from minimizing MSBE to minimizing the mean-square projected Bellman error (MSPBE). Although GTD2 can successfully enhance the learning *reliability*, it introduces an extra learning process and hence more complexity. Due to this reason, its use may potentially amplify negative interactions between the critic and the actor in ACRL and subsequently affect the learning effectiveness. This problem motivates us to consider another possible solution to improve the reliability in critic learning without bringing in extra learning complexity.

In typical ACRL algorithms, we find that the reliability of critic learning will deteriorate abruptly whenever the critic produces predicted cumulative rewards that fall outside the maximum/minimum possible cumulative rewards obtainable from any state in a learning environment. Such maximum/minimum possible cumulative rewards are problem-specific but can often be determined easily. Inspired by this finding, to improve the learning reliability, we decide to modify critic learning based on a Sandpile Model (SM) [9] to be explained in Section

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

II-C. SM has been frequently shown to enable a well-studied self-organized behavior in many systems [9]–[11]. We believe such a self-organized behavior can prevent critic learning from divergence, just like the SM can always maintain its criticality naturally.

A. Goals

Following the above idea, we conduct this research in two steps. Firstly, we introduce a criterion to measure the critic learning reliability by checking the learned value function along a sequence of previously visited states. Secondly, based on our reliability criterion, we propose an adapted SM which is further integrated with the critic learning process. As a result, the reliability of critic learning and the learned value function can be enhanced for effective RL.

Since these two steps are pretty general in nature, our adapted SM can be straightforwardly applied to many ACRL algorithms. However, in this research, we focus specifically on the *Regular Gradient Actor-Critic (RAC)* algorithm proposed by Bhatnagar et al. [12]. Consequently, a new algorithm variation featuring the use of our adapted SM is proposed and will be called the *Sandpile Model based Regular Gradient Actor-Critic (SMRAC)* algorithm. With SMRAC, we intend to answer two important research questions:

- Will the learning effectiveness of ACRL algorithms, such as RAC, be significantly improved upon using our adapted SM to maintain the critic learning reliability?
- To which extent will the effectiveness of RL and the critic learning reliability correlate with each other in ACRL algorithms, including RAC and SMRAC?

B. Organization

The remainder of the paper is organized as follows. Section II reviews the preliminary concepts for RL, the ACRL framework with the RAC algorithm, and the SM. Section III proposes a measurement for the critic learning reliability, adapts the SM to develop the SMRAC algorithm. Section IV discusses detailed experimental setups. Section V presents and analyzes the experimental results. The paper concludes in Section VI.

II. Preliminaries

This section introduces the preliminaries for our research and paves the way for the development of new ACRL algorithms. Firstly, the background of RL is presented. Secondly, the ACRL framework and the RAC algorithm are depicted. Thirdly, the concept of SM is explained.

A. Reinforcement Learning

The standard RL is a framework where an agent interacts with an unknown environment described by a Markov Decision Process (MDP) in the form of a 5-tuple model $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ [2]. $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to r \in \mathbb{R}$ is the reward function, and γ is a discount factor.

The goal of RL is to find an optimal policy that maximizes the expected cumulative rewards. In this work, we consider the stochastic policy, i.e., $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. For any policy π, we have the expected discounted cumulative rewards [1] defined as

$$J^{\pi} = V^{\pi}(\vec{s}_0) = \mathbf{E}_{\pi}[\sum_{t=0}^{T} \gamma^t r_{t+1} | \vec{s}_t = \vec{s}_0], \tag{1}$$

where V^{π} stands for the state-value function that gives the cumulative rewards obtained when the agent initiates from any state by following the policy π, and $r_{t+1} = \mathcal{R}(\vec{s}_t, a_t, \vec{s}_{t+1})$ is an instant scalar reward. Additionally, we can also have the expected cumulative rewards represented as an action-value function Q^{π}, i.e,

$$Q^{\pi}(\vec{s}, a) = \mathbf{E}_{\pi}[\sum_{t=0}^{T} \gamma^t r_{t+1} | \vec{s}_t = \vec{s}, a_t = a]. \tag{2}$$

Moreover, for any state $\vec{s}$, we can connect (1) with (2) as

$$V^{\pi}(\vec{s}) = \int_{a \in \mathcal{A}} \pi(\vec{s}, a) Q^{\pi}(\vec{s}, a) da. \tag{3}$$

Accordingly, the goal of identifying the optimal policy can be formulated as

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, V^{\pi}(\vec{s}_0). \tag{4}$$

It is worth noting that, this research focuses on episodic learning tasks where RL is conducted through multiple episodes. Specifically any single learning episode will be terminated when a terminal condition is satisfied, for example, the agent arrives at the goal region in a Puddle World or the maximum learning steps (T) is reached (see Section IV-A). For the simplicity of discussions, we use τ to denote an arbitrary episode.

B. Actor-Critic Reinforcement Learning Framework

In the literature, ACRL presents a important family of RL algorithms [4], [5]. Among them, our main research interest is on policy gradient based ACRL algorithms where actor learning is driven by policy gradients estimated by the learned critic. In other words, in order to achieve (4), all these algorithms attempt to update policy parameters through

$$\Delta \vec{\theta} \propto \nabla_{\vec{\theta}} J(\vec{\theta}), \tag{5}$$

in which

$$\nabla_{\vec{\theta}} J(\vec{\theta}) = \int_{\vec{s} \in \mathcal{S}} d^{\pi}(\vec{s}) \int_{a \in \mathcal{A}} \nabla_{\vec{\theta}} \pi_{\vec{\theta}}(a|\vec{s}) Q^{\pi}(\vec{s}, a) da d\vec{s}, \tag{6}$$

where $d^{\pi}(\vec{s}) = \lim_{t \to T} Pr\{\vec{s}_t = \vec{s} | \vec{s}_0, \pi\}$ denotes the stationary probability distribution of the states under π [13]. Despite of its simplicity, the policy gradient $\nabla_{\vec{\theta}} J(\vec{\theta})$ in (5) can only be estimated in reality.

In the literature, there are many possible ways to construct unbiased estimations of the policy gradient [12], [14], [15]. The most straightforward way is clearly demonstrated through

[1] In the following paper, we use "the expected cumulative rewards" to represent "the expected discounted cumulative rewards".

the *Regular Gradient Actor-Critic (RAC)* algorithm proposed in [12]. RAC is hence a simple and easily adaptable algorithm, particularly suitable for our research in this paper.

Critic learning in RAC is guided by the well-known Temporal Different error (TD error) defined as

$$\delta_t^{\pi} = r_{t+1} + \gamma V^{\pi}(\vec{s}_{t+1}) - V^{\pi}(\vec{s}_t). \tag{7}$$

Meanwhile the critic in RAC also approximates the true value function $V^{\pi}(\vec{s})$ in (7) by

$$V^{\pi}(\vec{s}) \approx \tilde{V}^{\pi}(\vec{s}) = \vec{\upsilon}^{\pi T} \cdot \phi(\vec{s}), \tag{8}$$

where $\vec{\upsilon}^{\pi T}$ consists of the value function parameters that are linearly associated with the basis function $\phi(\vec{s}) = [\phi_1(\vec{s}), \ldots, \phi_m(\vec{s})] \in \mathbb{R}^m$. Accordingly, the goal of critic learning is to adjust the value function parameters in the direction of reducing the TD error, i.e.,

$$\vec{\upsilon}_{t+1}^{\pi} \leftarrow \vec{\upsilon}_t^{\pi} + \alpha_t \delta_t^{\pi} \phi(\vec{s}_t), \tag{9}$$

where α_t is the critic learning rate at time t.

Actor learning in RAC aims to find the optimal policy parameter $\vec{\theta}^*$ so as to achieve (4). The learning follows the direction given by the policy gradient defined in (6) where Q^{π} is approximated as,

$$Q^{\pi}(\vec{s}, a) \approx \tilde{Q}^{\pi}(\vec{s}, a) = \vec{\omega}^{\pi T} \cdot \Phi(\vec{s}, a), \tag{10}$$

where $\vec{\omega}^{\pi}$ is made up of parameters that estimate $Q^{\pi}(\vec{s}, a)$, and the compatible feature is defined as $\Phi(\vec{s}, a) = \nabla_{\vec{\theta}} \ln \pi(\vec{s}, a)$. Subsequently, through an unbiased estimation of $\nabla_{\vec{\theta}} J(\vec{\theta})$ in (6), the incremental rule for actor learning is further proposed in [12] as

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \beta_t \delta_t^{\pi} \Phi(\vec{s}, a), \tag{11}$$

where β_t is the actor learning rate.

Through iterative application of (9) and (11), RAC is widely shown to successfully solve many benchmark RL problems. It is clear to see that in RAC the estimation of the policy gradients in (11) is strongly dependent on the quality of critic learning. Also note that, Algorithm 1 can be regarded as the exact algorithmic description of RAC after excluding lines from 14 to 28 as indicated by horizontal lines.

C. Sandpile Model

The SM (a.k.a., Bak-Tang-Wiesenfeld model) is a model proposed in [9] to explain an important property of dynamical systems called *Self Organized Criticality (SOC)*. SOC refers to a phenomenon that a system evolves towards a critical point through self-adjustments over its lifetime. In SM, whenever a system (e.g., a sand pile) reaches its critical point, any small perturbation (e.g., dropping a grain of sand) may trigger a *noise* propagation (e.g., an avalanche of the sand pile) of varying sizes. After a short while, the system is guaranteed to return back to its the critical point through self-organization.

In [9], the SM is simulated in a 2D $N \times N$ grid with a boundary (i.e., $[0, N]^2$), where each cell (i.e., site (x, y)) of the grid contains an integer value $z(x, y)$ representing the slope of the sand pile on-site. It is also assumed that, if a grain of sand is added one at a time on a random site and it leads to an avalanche, then only after the avalanche stops can the next grain of sand be added. Following this setting, SM as a mathematical model contains three key components:

- *Neighborhoods and Boundaries*
  In the 2D SM, the neighborhoods of any site (x, y) are defined as its four adjoining sites, namely $(x \pm 1, y)$ and $(x, y \pm 1)$. Besides, the sites on boundaries are defined as $(0, y)$, (N, y), $(x, 0)$, and (x, N).
- *The Reliability Criterion*
  A key factor of the SM is to determine the condition under which an avalanche will be triggered upon adding a new grain of sand. This is equivalent to defining a reliability criterion as below,

  $$\varepsilon(z) = \begin{cases} 0, & K - z(x, y) \geq 0 \\ 1, & K - z(x, y) < 0 \end{cases} \tag{12}$$

  where K is the predefined threshold (the critical value). As seen in (12), the site is reliable when its z value is smaller than K, i.e., $\varepsilon(z) = 0$. On the other hand, an avalanche will be triggered at any site (x, y) when the value $z(x, y)$ exceeds K.
- *The Propagation/Updating Rule*
  An avalanche causes a noise propagation to its neighborhoods, such a propagation can be defined as

  $$\begin{aligned} z(x, y) &\leftarrow z(x, y) - \Delta \\ z(x \pm 1, y) &\leftarrow z(x \pm 1, y) + \tfrac{\Delta}{4} \\ z(x, y \pm 1) &\leftarrow z(x, y \pm 1) + \tfrac{\Delta}{4} \end{aligned}\ , \tag{13}$$

  where Δ represents the noise to be propagated from the site (x, y) to its neighborhoods. Note that, the noise being propagated to the boundaries will be disregarded, i.e., $z(x, y) \equiv 0$.

Clearly, although the model description above considers only two dimensions, the SM can be easily expanded to multiple dimensional cases [9].

III. A Sandpile Model based Actor-Critic Reinforcement Learning Algorithm

In this section, we propose the SMRAC algorithm with the aim of improving the critic learning reliability. For this purpose, we firstly propose a criterion for measuring the learning reliability. Following that, we show how to adapt the SM for the purpose of combining it with the critic learning process in RAC. Lastly, we present an algorithmic description of SMRAC.

A. Critic Learning Reliability

We define the concept of the critic learning reliability below.

Definition III.1. The *Critic Learning Reliability* refers to the total probability for the absolute value generated by the learned value function to be greater than a predefined threshold across all previously visited states in the current episode.

Since $\vec{\upsilon}$ and $\vec{\theta}$ jointly determine the behavior of RAC, the reliability criterion for critic learning during any learning episode τ can be presented mathematically as

$$\varepsilon_\tau(\vec{\upsilon},\vec{\theta}) = \int_{\vec{s}\sim d^\pi(\vec{s})} \mathbf{I}(\vec{s}) d^\pi(\vec{s}) d\vec{s}, \tag{14}$$

where

$$\mathbf{I}(\vec{s}) = \begin{cases} 0, & \bar{R} - |\vec{\upsilon}^T \cdot \phi(\vec{s})| \geq 0 \\ 1, & \bar{R} - |\vec{\upsilon}^T \cdot \phi(\vec{s})| < 0 \end{cases}, \tag{15}$$

is an indicator function based on a predefined threshold $\bar{R}$. The choices of $\bar{R}$ for practical learning tasks will be explained in Section IV-B3.

ε in (14) is a continuous measure of critic learning reliability. In particular, $\varepsilon_\tau(\vec{\upsilon},\vec{\theta}) = 0$ indicates that the critic learning is *completely reliable*. Otherwise, $\varepsilon_\tau(\vec{\upsilon},\vec{\theta}) > 0$ shows the degree of divergence in the learned value function.

B. An Adapted Sandpile Model for Critic Learning

To integrate the SM into RAC, based on the three components of the SM described in Section II-C, we propose the adapted SM for critic learning as follows.

- Neighborhoods and Boundaries
  Our adapted SM is defined over the history $\mathcal{H}$ of all previously visited states in an episode. In particular, each state (and its corresponding value) is treated as a separate site in the SM. Hence, any state $V^\pi(\vec{s}_j)$ has its neighborhoods made up of $V^\pi(\vec{s}_{j-1})$ and $V^\pi(\vec{s}_{j+1})$. Moreover, the boundary of the SM is determined by the most recently and least recently visited states in the history, whereas the length of history is bounded from above by $l_{\mathcal{H}}$.
- *The Reliability Criterion*
  Referring to the reliability criterion in Section III-A, the critical point value K in (12) for our criterion is defined as the upper bound of the expected cumulative reward, i.e., $\bar{R}$. Intuitively whenever the critic in RAC predicts non-achievable cumulative rewards, falling outside the range defined by $\bar{R}$, it is highly skeptical that critic learning starts to become unreliable. Based on this idea, we can define the absolute bound $\bar{R}$ as,

  $$\bar{R} = \sum_{i=0}^{T} \gamma^i |r_i| + \epsilon, \tag{16}$$

  where ϵ is a small *error margin* which is necessary since the value function is estimated by linear function approximation in RAC.
- *The Propagation/Updating Rule*
  Suppose that at least one site, i.e., $\vec{s}_j$, over the full history has been identified as unreliable, similar to the original SM in [9], the propagation/updating rule described below can be applied:

  $$\begin{array}{rl} \tilde{V}^\pi_{t+1}(\vec{s}_j) & \leftarrow \tilde{V}^\pi_{t+1}(\vec{s}_j) - \Delta \\ \tilde{V}^\pi_{t+1}(\vec{s}_{j-1}) & \leftarrow \tilde{V}^\pi_{t+1}(\vec{s}_{j-1}) + \rho\frac{\Delta}{2} \\ \tilde{V}^\pi_{t+1}(\vec{s}_{j+1}) & \leftarrow \tilde{V}^\pi_{t+1}(\vec{s}_{j+1}) + \rho\frac{\Delta}{2} \end{array} . \tag{17}$$

  Δ in (17) is defined below

  $$\Delta \; = \mathrm{sign}(\tilde{V}^\pi(\vec{s}_t))\psi_{\bar{R}}\bar{R} \;, \tag{18}$$

  where $\mathrm{sign}(\tilde{V}^\pi(\vec{s}_t)) = \begin{cases} 1, & \tilde{V}^\pi(\vec{s}_t) \geq 0 \\ -1, & \tilde{V}^\pi(\vec{s}_t) < 0 \end{cases}$.
  Note that, $\psi_{\bar{R}} \in [0,1]$ is the noisy level factor. For example, if $\psi_{\bar{R}} = 0.1$, it means that 90% of the maximum allowed cumulative rewards will be maintained.
  In (17), we define a new meta parameter $\rho \in [0,1]$ as the damping factor to avoid big changes to the critic that can potentially affect the learning effectiveness of RAC. Moreover, based on (8), we can have (17) re-written as

  $$\begin{array}{rl} \vec{\upsilon}_{t+1} & \leftarrow \vec{\upsilon}_{t+1} - \frac{\Delta}{\phi(\vec{s}_j)} \\ \vec{\upsilon}_{t+1} & \leftarrow \vec{\upsilon}_{t+1} + \frac{0.5\rho\Delta}{\phi(\vec{s}_{j-1})} \\ \vec{\upsilon}_{t+1} & \leftarrow \vec{\upsilon}_{t+1} + \frac{0.5\rho\Delta}{\phi(\vec{s}_{j+1})} \end{array} . \tag{19}$$

The adapted SM proposed above will be applied iteratively at every learning step so as to guarantee critic learning reliability.

C. The SMRAC Algorithm

Our adapted SM stated above can be easily incorporated into the RAC algorithm. First, we maintain at most $l_{\mathcal{H}}$ recently visited states in history. Next, the reliability of the SM over all visited states is examined according to (14). Subsequently, if there exists one or multiple unreliable sites, one of them will be randomly selected and the propagation rule (19) will be applied to it. This procedure will be repeated until all sites in the SM are reliable. We present an algorithmic description of SMRAC in Algorithm 1.

IV. Experiment Design

To answer the two research questions proposed in Section I, we have performed experiments on two benchmark problems. The section starts with an introduction to the two problems. Next, we discuss the detailed experimental setups.

A. Benchmark Problems

In this research, we choose two continuous RL problems, namely the Puddle World problem and the Mountain Car problem.We choose these two problems because: 1) They have already been widely used for evaluating many ACRL algorithms. 2) The threshold on obtainable cumulative rewards in (16) can be easily determined on both problems. 3) They are sufficiently simple to help us study the impact of critic learning reliability on the effectiveness of ACRL algorithms.

1) Puddle World Problem [2]: The Puddle World problem is a two-dimensional continuous environment (i.e., $[0,1]^2$) in which round puddles are placed at (0.2, 0.25) to (0.55, 0.25) and (0.45, 0.2) to (0.45, 0.6) with a radius 0.1. A mobile agent initiates at a random position in the environment and learns to reach the goal region (i.e., $x + y \geq 1.9$) without entering the puddles. When reaching the goal region, the agent will receive an instant reward of "+40". Otherwise, it will be penalized with "-1" for its movement. In particular, when entering the puddle area, it receives a penalty computed by

Algorithm 1 The SMRAC Algorithm

```
Require: an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, the expected reward upper bound $\bar{R}$, the noisy level factor $\psi_{\bar{R}}$, the damping factor $\rho$, the maximum length for the state history $l_{\mathcal{H}}$
Ensure: $\vec{\theta}$, $\vec{v}^{\pi}$
    Initialization:
    $\vec{\theta} \leftarrow \vec{\theta}_0$
    $\vec{v}^{\pi} \leftarrow \vec{v}^{\pi}_0$
    $\vec{s}_t \leftarrow \vec{s}_0$, where $\vec{s}_0$ is an arbitrary initial state
    $\mathcal{H} \leftarrow \{\}$
    $l \leftarrow 0$
    $j \leftarrow 0$
    $\Delta \leftarrow 0$
    Learning Process:
for $\tau = 0, 1, 2, \ldots, \tau_{max}$ do
    for $t = 0, 1, 2, \ldots, T$ do
        $a_t \sim \pi_{\vec{\theta}}(a|\vec{s}_t)$
        Take action $a_t$, observe reward $r_{t+1}$ and new state $\vec{s}_{t+1}$
        $\delta^{\pi}_t \leftarrow r_{t+1} + \gamma \vec{v}^{\pi T}_t \cdot \phi(\vec{s}_{t+1}) - \vec{v}^{\pi T}_t \cdot \phi(\vec{s}_t)$
        $\vec{v}^{\pi}_{t+1} \leftarrow \vec{v}^{\pi}_t + \alpha_t \delta^{\pi}_t \phi(\vec{s}_t)$
        ----------------------------------------
        $\mathcal{H} \leftarrow \mathcal{H} \cup \{\vec{s}_t\}$
        if $l < l_{\mathcal{H}}$ then
            $l \leftarrow l + 1$
        else
            $\mathcal{H} \leftarrow \mathcal{H} \setminus \vec{s}_{t-l}$
        while $\varepsilon_{\tau}(\vec{v}_{t+1}, \vec{\theta}_t) > 0$ do
            while $\vec{s}_j \in \mathcal{H}$ do
                if $\bar{R} < |\vec{v}^{T}_{t+1} \cdot \phi(\vec{s}_j)|$ then
                    $\Delta \leftarrow \text{sign}(\vec{v}^{\pi T}_{t+1} \cdot \phi(\vec{s}_j))\bar{R}\psi_{\bar{R}}$
                    $\vec{v}_{t+1} \leftarrow (1 - \frac{\Delta}{\vec{v}_{t+1} \cdot \phi(\vec{s}_j)})\vec{v}_{t+1}$
                    if $j > 1$ then
                        $\vec{v}_{t+1} \leftarrow (1 + \frac{0.5\rho\Delta}{\vec{v}_{t+1} \cdot \phi(\vec{s}_{j-1})})\vec{v}_{t+1}$
                    if $j < l_{\mathcal{H}} - 1$ then
                        $\vec{v}_{t+1} \leftarrow (1 + \frac{0.5\rho\Delta}{\vec{v}_{t+1} \cdot \phi(\vec{s}_{j+1})})\vec{v}_{t+1}$
                $j \leftarrow j + 1$
        ----------------------------------------
        $\vec{\theta}_{t+1} \leftarrow \vec{\theta}_t + \beta_t \delta^{\pi}_t \Phi(\vec{s}_t, a_t)$
    $\mathcal{H} \leftarrow \{\}$
    $l \leftarrow 0$
    $j \leftarrow 0$
    $\Delta \leftarrow 0$
return $\vec{\theta}$, $\vec{v}^{\pi}$
```

multiplying "-400" with the agent's shortest distance to the border of the puddle [16]. The agent moves in the environment with a direction (i.e., an action) $a \in [-\pi, \pi]$ and a speed of 0.05, which is modeled as

$$\begin{aligned} x_{t+1} &= x_t + 0.05 \times sin(a) \\ y_{t+1} &= y_t + 0.05 \times cos(a) \end{aligned}.$$

In this problem, a learning episode is defined as the learning period from the agent's initial state to the termination moment when it arrives at the goal region.

2) Mountain Car Problem [2]: The Mountain Car problem is an environment with two-dimensional state input including the position of the car (i.e., $x \in [-0.5, 0.5]$) and the velocity of the car (i.e., $\dot{x} \in [-0.08, 0.08]$). The problem is to drive an underpowered car in a valley up a steep mountain at the right-hand side. Due to the difficulty of that the gravity is stronger than the car's engine, the car can only climb the mountain with the aid of descending acceleration obtained by reversing itself up to the opposite slope of the left-hand side. A goal region (i.e., $x \geq 0.5$) is defined on the top of the mountain, and the aim of the problem is for the car to reach the goal region in the minimum steps. Every movement of the car in the environment receives a penalty "-1", a reward "+10 " is given when the car reaches the goal region. The environment dynamics for the car to update its location and its speed is modeled as

$$\ddot{x} = \dot{a}\mathcal{F} - 0.0025\ \cos(3x),$$

where $\dot{a} = 0.001$ is the acceleration of the car, and $\mathcal{F} \in [-10.0, 10.0]$ is the force (i.e., action) performed by the system.

Similar to the Puddle World problem, one learning episode of the Mountain Car problem terminates when either the goal region or the maximum steps T is reached.

B. Experiment Setup

In this subsection, we describe the detailed setups of our experiments. The general experiment running setups are firstly described. We then give the formulation of the stochastic policy (i.e., Gaussian Policy) used in this research. Subsequently, we discuss the basis function as well as some important meta parameters settings adopted by both RAC and SMRAC..

To obtain reliable experimental results, for each learning algorithm and each benchmark problem, we will perform 50 independent trials (i.e., one complete training and testing process) over 10000 training episodes. For one trial, after every 50 training episodes, we will run 50 tests to verify the current performance of the actor learned by RAC and SMRAC respectively. Furthermore, a maximum number of 100 steps applies to every training and testing episode. In addition to these settings, please find other implementation details below.

1) Stochastic Policy Implementation: We implement our stochastic policy as a Gaussian distribution which is well-studied for continuous problems [17]. Specifically, the probability density for taking each action is given by,

$$\pi_{\vec{\theta}}(a|\vec{s}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(a-\mu)^2}{2\sigma^2}},$$

where μ is determined by the policy parameters $\vec{\theta}$ and the basis function $\phi(\vec{s})$, i.e., $\mu = \vec{\theta}^T \cdot \phi(\vec{s})$. Meanwhile, σ is considered as an exploration parameter and is fixed to 1.0 for all problems. Note that, π at the RHS of (20) is the circumference ratio.

2) Basis Function: In this work, we use the triangle basis function presented in [16] to project the low-level state inputs to the high-level feature spaces, i.e.,

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^m,$$

where d is the dimension of the state input, and m is the feature dimension.

Fig. 1 depicts how the triangle basis function is used for one dimension of the state input. As seen in the figure, the dimension of the state input is limited in the range interval $[\iota_{min}, \iota_{max}]$. We then split the interval into n equal segments with $n-1$ vertexes, for this case, $n = 10$. Next, we select each vertex as an apex to connect with the adjacent vertexes to construct a group of triangles. For the cases when the adjacent vertexes are ι_{min} or ι_{max}, we connect the apex to the boundary instead of vertexes. For each triangle, it covers a partial range of the state input. When the value of the dimension s_d for the incoming state input (i.e., $\vec{s} = [s_0, s_1, \ldots, s_d] \in \mathbb{R}^d$) falls into the range, its corresponding features will be computed as,

$$\phi(s_k)_i = \begin{cases} 1, & \iota_{min} \leq s_k < \iota_{min} + \kappa \\ 1, & \iota_{max} - \kappa < s_k \leq \iota_{max} \\ \frac{s_k - \kappa*(i+1)}{\kappa*(i+1) - \kappa*i} + 1, & s_k \leq \iota_{min} + \kappa*(i+1) \\ \frac{s_k - \kappa*(i+2)}{\kappa*(i+2) - \kappa*(i+1)}, & s_k \geq \iota_{min} + \kappa*(i+1) \\ 0, & \text{otherwise} \end{cases},$$

where $k \leq d$, and $i = 0, 1, \ldots, n-1$, meanwhile $\kappa = \frac{|\iota_{min}| + |\iota_{max}|}{n}$. Note that, the final feature dimension is determined as $m = d * n$.

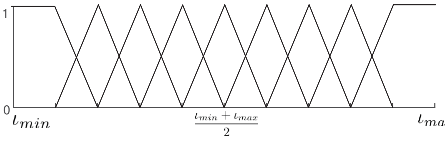

Fig. 1: The triangle basis function used for defining one single dimension of the state input.

3) Meta Parameter Settings: To investigate the impact of the critic learning reliability on the learning performance, we followed those meta parameter settings summarized in Table I. They enable us to study the behavior of RAC when critic learning is clearly unreliable or even diverging. This is important because, when the learned value function satisfies our reliability criterion, SMRAC and RAC behave exactly the same.

Algorithms	Problems	Meta Parameters						
		α	β	γ	$\bar{R}$	$\psi_{\bar{R}}$	ρ	$l_{\mathcal{H}}$
RAC	Puddle World	0.01	0.0001	0.99	N/A	N/A	N/A	N/A
	Mountain Car	0.1	0.005	0.99	N/A	N/A	N/A	N/A
SMRAC	Puddle World	0.01	0.0001	0.99	120	0.1	0.1	100
	Mountain Car	0.1	0.005	0.99	100	0.1	0.1	100

TABLE I: The meta-parameter settings for experiments of RAC and SMRAC on the Puddle World problem and the Mountain Car problem.

The settings for $\bar{R}$ as seen in Table I are problem-specific. Regarding the Puddle World problem, following the reward scheme described in Section IV-A1 and (16), the maximum expected cumulative reward $\bar{R}$ is determined as 120. We can easily think that the worst case is that the agent is trapped in the environment obtaining "-1.0" mostly for 100 steps so that we can have theoretical maximum cumulative rewards as "100". Additionally, the error margin (i.e., ϵ) here is set to 20% of the maximum theoretical value. Similarly, in the Mountain Car problem, the threshold $\bar{R}$ can be determined as "100".

V. Results and Discussion

In this section, we present and analyze the experimental results. We will discuss the experimental results on two benchmark problems separately. For the discussion on results collected from each problem, we will firstly compare the critic learning reliability of RAC and SMRAC. Based on the reliability differences, we further compare the learning performance of the two algorithms. Lastly, we investigate the relationship between the learning effectiveness and the learning reliability by adopting a correlation analysis.

A. Experimental Results on Puddle World

To evaluate the critic learning reliability, we present the average of the absolute values generated from the value function learned by RAC and SMRAC at every 50 episodes in Fig. 2. As seen in Fig. 2, the RAC algorithm exhibits a very unreliable behavior since its corresponding learned critic values fluctuate severely. A sudden change occurs at the 1350-th episode indicated as a black dashed line, then it tends to diverge rapidly. Such a change results in an immediate degradation in the learning performance in terms of the average cumulative rewards as shown in Fig. 3. In contrast, also from Fig. 2, the critic learning reliability of SMRAC has been well maintained at a reasonable level, staying mostly beneath the predefined threshold $\bar{R}$. Correspondingly, the learning performance of SMRAC also shows a converging behavior evidenced in Fig. 3.

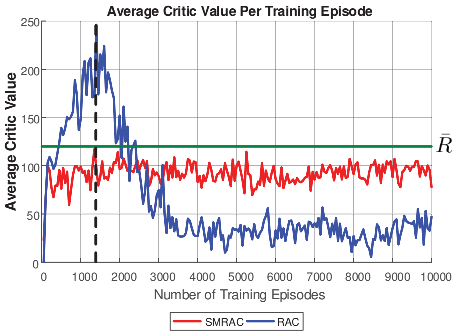

Fig. 2: Average of the absolute values generated from the value function learned by RAC and SMRAC at every 50 episodes on the Puddle World problem.

Next, we will compare the learning performance of SMRAC and RAC based on average cumulative rewards in Fig. 3. As discussed above, after the 1350-th episode, the entire learning process of RAC diverges and never recovers due to its unreliable critic learning. In contrast, SMRAC shows a continuous improvement on the learning performance, although it tends to converge after 1350-th episode thanks to the convergence of its critic learning. Additionally, we have performed a Student T-test for comparing the performances of the two algorithms which produces a p-value of 9.6864×10^{-10}. This suggests that SMRAC performs significantly better than RAC on the Puddle World problem.

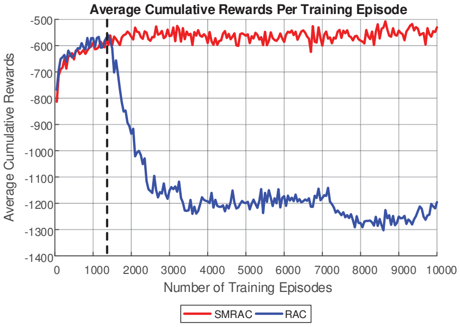

Fig. 3: Average cumulative rewards obtained by RAC and SMRAC at every 50 episodes on the Puddle World problem.

In summary, Fig. 2 and Fig. 3 reflect some facts that 1) SMRAC performs learning more reliably and more effectively than RAC does, and 2) the learning reliability to some extent correlates to the learning effectiveness.

Aiming at knowing to what extent the critic learning reliability affects the learning effectiveness, we also perform a correlation analysis. As seen from Fig. 2 and Fig. 3, it is easy to observe that there exists a correlation between the learning reliability and the learning effectiveness. To verify this correlation, we adopt the Pearson Correlation analysis here. Firstly, we follow (14) to quantify the learning reliability. In addition, we follow the convention to use average cumulative rewards to measure the learning effectiveness. The correlation matrix for RAC on the Puddle World problem is given in Fig. 4, which shows a correlation coefficient of -0.87 close to -1. This suggests that the learning reliability has a strong positive correlation because higher value in (14) indicates poorer reliability for critic learning to the learning performance.

B. Experimental Results on Mountain Car

In comparison to the Puddle World problem, very similar experimental results can be obtained on the Mountain Car problem. Accordingly, most of the claims made in Section V-A also hold here.

Fig. 5 shows the average value of the learned critic by RAC and SMRAC on the Mountain Car problem. Clearly, the critic learning of RAC diverges quickly after 450 episodes and the value of critic keeps increasing until the 3650-th episode. Even after the critic value starts to decrease at the 3650-th episode, it cannot prevent policy parameters from diverging further. In contrast, a clear converging trend of SMRAC can be witnessed in Fig. 5 towards the critical point $\bar{R}$, suggesting the higher reliability of SMRAC compared to that of RAC.

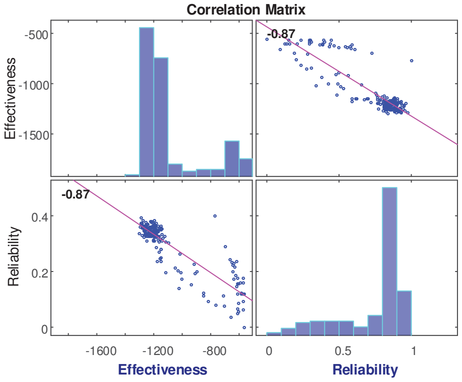

Fig. 4: Correlation between the learning effectiveness and the learning reliability of RAC on the Puddle World problem.

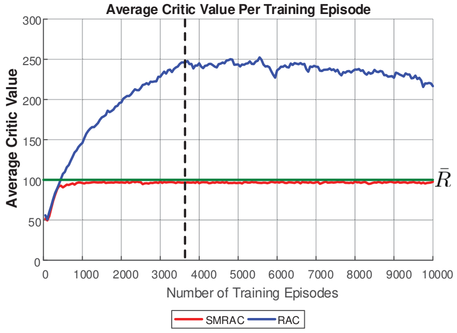

Fig. 5: Average of the absolute values generated from the value function learned by RAC and SMRAC at every 50 episodes on the Mountain Car problem.

As in Fig. 6, the evident difference can be easily identified the fact that SMRAC performs significantly better than RAC. This fact is supported by the significance test where the p-value is 0.0145.

The correlation analysis based on the results collected by RAC on the Mountain Car problem is presented in Fig. 7. This analysis also indicates that a relatively strong positive correlation exists between the learning reliability and the learning effectiveness of RAC, as demonstrated by the correlation coefficient equals to -0.74.

C. Summary

By conducting the above experiments, we can confidently answer the two research questions presented in Section I. In regard to the first research question, we found that SMRAC

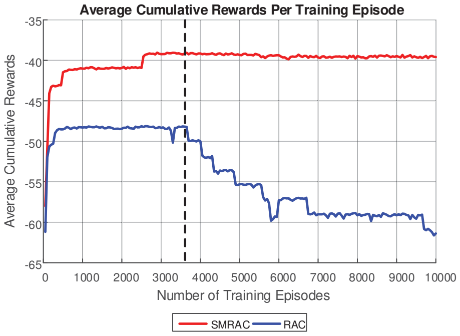

Fig. 6: Average value function learned by RAC and SMRAC at every 50 episodes on the Mountain Car problem.

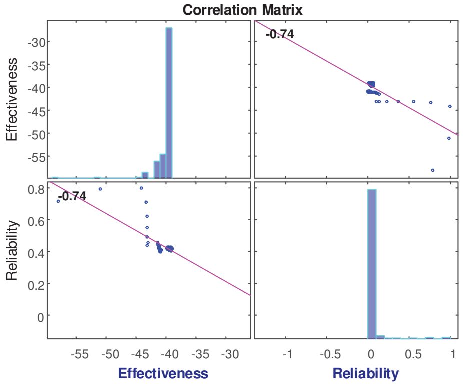

Fig. 7: Correlation between the learning effectiveness and the learning reliability of RAC on the Mountain Car problem.

outperformed RAC on both benchmark problems significantly, yet neither algorithm has obtained the theoretical best performances shown in [16]. In fact, to achieve the best performance, a fine-tuning process for meta parameters must be conducted. To answer the second question, we have adopted the correlation analysis on the RAC algorithm on both benchmark problems. The correlation coefficients are respectively -0.87 and -0.74, both support that the relatively strong positive correlation [18] holds between learning reliability and learning performance.

VI. Conclusions

In this paper, we investigated in-depth the use of the sandpile model for improving critic learning reliability in ACRL algorithms. Driven by the understanding that critic learning will become unreliable whenever the critic predicts non-achievable cumulative rewards over the entire history of previously visited states, we have further proposed a new ACRL algorithm called SMRAC. SMRAC was designed based on RAC, and it featured the use of an adapted SM for critic learning. Experimental evidence clearly showed that SMRAC outperformed RAC in terms of learning reliability as well as effectiveness on two benchmark problems. Moreover, our experimental results also demonstrated that learning reliability and learning effectiveness are strongly correlated. These findings potentially provide a new direction for the development of more useful RL algorithms. In the near future, we plan to apply the adapted SM to a wide-range of ACRL algorithms with evaluations on more RL problems to more comprehensively assess its practical usefulness.

Acknowledgment

The authors would like to thank the anonymous IJCNN reviewers for insightful comments, and the New Zealand eScience Infrastructure organization providing High-Performance Computing resources to facilitate this research.

References

[1] M. L. Littman, “Reinforcement learning improves behaviour from evaluative feedback.” *Nature*, vol. 521, no. 7553, pp. 445–51, 2015.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning : An Introduction*, 1998.

[3] M. Geist and O. Pietquin, “Algorithmic survey of parametric value function approximation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 845–867, 2013.

[4] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuška, “A survey of actor-critic reinforcement learning: Standard and natural policy gradients,” pp. 1291–1307, 2012.

[5] V. R. Konda and J. N. Tsitsiklis, “Actor-Critic Algorithms,” *Control Optim*, vol. 42, no. 4, pp. 1143–1166, 2003.

[6] ——, “Convergence rate of linear two-time-scale stochastic approximation,” *Annals of Applied Probability*, vol. 14, no. 2, pp. 796–819, 2004.

[7] R. S. Sutton, C. Szepesvari, and H. R. Maei, “A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 21, pp. 1609–1616, 2009.

[8] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, E. Wiewiora, and C. Szepesvari, “Fast Gradient-descent Methods for Temporal-difference Learning with Linear Function Approximation,” *Proceedings of the 26th Annual International Conference on Machine Learning*, vol. 1, no. 7, pp. 993–1000, 2009.

[9] P. Bak, C. Tang, and K. Wiesenfeld, “Self-organized criticality: An explanation of the 1/f noise,” *Physical Review Letters*, vol. 59, no. 4, pp. 381–384, 1987.

[10] É. Goles, M. Latapy, C. Magnien, M. Morvan, and H. D. Phan, “Sandpile models and lattices: A comprehensive survey,” *Theoretical Computer Science*, vol. 322, no. 2, pp. 383–407, 2004.

[11] D. Dhar, “The Abelian Sandpile and Related Models,” *Physica A*, vol. 263, pp. 4–25, 1999.

[12] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, “Natural actor-critic algorithms,” *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[13] R. S. Sutton, D. Mcallester, S. Singh, and Y. Mansour, “Policy Gradient Methods for Reinforcement Learning with Function Approximation,” *Advances in Neural Information Processing Systems 12*, pp. 1057–1063, 1999.

[14] J. Peters and S. Schaal, “Policy gradient methods for robotics,” in *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 2219–2225.

[15] M. P. Deisenroth, G. Neumann, J. Peters, and N. Publishers, *A Survey on Policy Search for Robotics*, ser. Foundations and trends in robotics, 2013. [Online]. Available: https://books.google.co.nz/books?id=hPLGoQEACAAJ

[16] G. Chen, C. I. J. Douch, and M. Zhang, “Reinforcement Learning in Continuous Spaces by using Learning Fuzzy Classifier Systems,” *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, p. 1, 2016.

[17] J. Peters and S. Schaal, “Policy gradient methods for robotics,” in *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 2219–2225.

[18] W. L. Hays, “An Introduction to Linear Regression and Correlation. 2nd ed.” *PsycCRITIQUES*, vol. 30, no. 10, 1985.