The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

# R$^2$PG: Risk-Sensitive and Reliable Policy Gradient

**Bo Liu,**[1] **Ji Liu,**[2] **Kenan Xiao**[1]
[1]Department of Computer Science and Software Engineering, Auburn University, AL
[2]Department of Computer Sciences, University of Rochester, NY
boliu@auburn.edu, jliu@cs.rochester.edu, kzx0010@auburn.edu

### Abstract

Policy gradient approaches have gained great success in many complex dynamic decision-making problems, such as the game of Go. However, policy gradient methods suffer from high variance, which implies weak risk control in real applications. Therefore, it is valuable to introduce variance reduction techniques into policy gradient methods to help control the variance in the policy improvement process. Meanwhile, risk-sensitive management in dynamic decision problems is a primary concern in many fields, such as finance and process control. In this paper, we developed a policy search framework for reinforcement learning with variance-related criteria and a variance reduction technique. Our starting point is a standard formulation for the variance of the cost-to-go in episodic tasks. Using this formula, variance-reduced policy search algorithms are proposed. The convergence to local optima of the proposed algorithms is proved, and their applicability is demonstrated on financial-portfolio domains.

## 1 Introduction

Policy gradient methods, which originate from the REINFORCE algorithm (Williams 1992), is a family of stochastic gradient policy improvement methods based on the likelihood ratio trick (Williams 1992). Such family of algorithms is known to be on-policy stable, yet suffers from a huge variance. Two techniques have been widely used to reduce the variance, i.e., the baseline method (Peters and Schaal 2006) and the actor-critic method (Greensmith, Bartlett, and Baxter 2004).Both can be interpreted as additive control variate methods aim at reducing the variance of the learning process.

To reduce variance, only controlling the variance in the learning process may not be enough. Another widely used technique is to use risk-sensitive objective functions instead of the vanilla objective function, which only considers maximizing the expected sum of return. Such kind of methods are widely used in risk management, robust decision-making, etc. (Chow and Ghavamzadeh 2014; Tamar, Castro, and Mannor 2012). The risk is usually depicted by the variance of the expected sum of return. The basic motivation of risk-sensitive approaches, therefore, is to consider the variance control term in the objective function. The mean-variance trade-off objective function, which is a standard risk-sensitive objective function, is widely used in the risk-sensitive policy improvement approaches. However, the regular likelihood-ratio based stochastic policy gradient methods can not apply directly to solve such objective functions, as pointed out by (Prashanth and Ghavamzadeh 2013; Tamar, Castro, and Mannor 2012). The major reason is similar to the double sampling problem, e.g., the variances can not be accurately estimated from a single trajectory. Recently, a stochastic composition optimization framework is proposed for such problems (Wang, Fang, and Liu 2017), which demonstrates great potential in convergence rate acceleration (Wang, Liu, and Fang 2016) and variance reduction (Lian, Wang, and Liu 2016).

This paper is motivated by two factors. First, to the best of our knowledge, there is little research that focuses on both the risk-awareness and reliability of policy gradient algorithms, i.e., both a risk-sensitive objective function and the variance reduction technique are used. Secondly, stochastic variance reduction methods, such as the SVRG method (Johnson and Zhang 2013),have shown promising performances in stochastic optimization problems. It is intriguing to apply such stochastic variance reduction techniques to policy gradient algorithms as a counterpart to the existing control variate based approaches. Motivated by these two factors, we first extend the stochastic composition gradient method with variance reduction method to non-convex problems, then propose a novel policy gradient algorithm which takes both the risk-sensitivity and learning reliability into consideration.

Here is a road map to the rest of the paper. In Section 2, the background on stochastic variance reduction is introduced. In Section 3, two novel algorithms are proposed based on the stochastic variance reduction techniques. A detailed experimental study in Section 4 validates the effectiveness of the proposed algorithm.

## 2 Stochastic Variance Reduction

*Stochastic variance reduction* technique is one of most notable achievements in stochastic optimization. Two most widely used methods, SVRG (Johnson and Zhang 2013), and SAGA (Defazio, Bach, and Lacoste-Julien 2014), achieve the same low computational cost per iteration as well as a fast, linear convergence rate compared with conventional stochastic gradient approaches by utilizing the structure of the problem. Both of these two approaches require the objective function $f(x)$ to be a well-defined strongly convex function with a finite-sum structure, i.e.,

$$f(x) = \sum_{i=1}^{n} F_i(x). \tag{2.1}$$

Furthermore, there are several attempts to extend SVRG to other problems, such as non-strongly-convex and sum of non-convex problems(Zhu and Yuan 2016), non-convex problems (Reddi et al. 2016) (Zhu and Hazan 2016), Riemannian SVRG (Zhang, Reddi, and Sra 2016), and saddle-point problems (Palaniappan and Bach 2016). The SVRG algorithm has a nested loop structure. In the outer loop, it stores a reference points $\tilde{x}$, which is updated at the outer loop,and remains unchanged in the inner loop iterations of the algorithm. In the inner loop, the algorithm computes $\nabla f\left(\tilde{x}\right) = 1/n \sum\limits_{i=1}^{n} \nabla F_i(\tilde{x})$ at the reference point $\tilde{x}$. It estimates the gradient at each iteration based on the reference gradient

$$\nabla \tilde{f}\left(x_k\right) = \nabla F_{i_k}(x_k) - \nabla F_{i_k}(\tilde{x}) + \nabla f\left(\tilde{x}\right), \tag{2.2}$$

where $i_k$ is uniformly randomly sampled from $\{1, 2, \cdots, n\}$ at $k$-th inner iteration, k is the index of inner loop. The key improvement of SVRG is that the variance $E[\|\nabla \tilde{f}(x_k) - \nabla f(x_k)\|^2]$ decreases to zero as $x_k$ converges to the optimal. This variance reduction also uses the fixed step that has the linear convergence rather than the decreasing step in SGD to sub-linear convergence.

### 2.1 Stochastic Composite Gradient Variance Reduction

The stochastic composite gradient variance reduction is proposed by Lian *et.al.* (Lian, Wang, and Liu 2016), which considers the finite-sum scenario for composition optimization:

$$\min_x f\left(x\right) = \frac{1}{n}\sum_{i=1}^{n} F_i(\frac{1}{m}\sum_{j=1}^{m} G_j\left(x\right)). \tag{2.3}$$

We define

$$G\left(x\right) = \frac{1}{m}\sum_{j=1}^{m} G_j\left(x\right), F\left(G\left(x\right)\right) = \frac{1}{n}\sum_{i=1}^{n} F_i\left(G\left(x\right)\right). \tag{2.4}$$

In the compositional-SVRG algorithm, there are two aspects using variance reduction technique. The first one is similar to SVRG method but instead of estimating the gradient, it estimates $G(x)$ through,

$$\hat{G}_k = \frac{1}{A}\sum_{1\leq j\leq A}\left(G_{A_k[j]}\left(x_k\right) - G_{A_k[j]}\left(\tilde{x}_v\right)\right) + G\left(\tilde{x}_v\right), \tag{2.5}$$

where $\tilde{x}_v$ is current outer iteration, $x_k$ is the current iteration. $A$ is the sampling times to form the mini-batch multiset $\mathcal{A}_k$, which is the key operation in analyzing the query complexity of the non-convex composition problem. The second one is also similar to the SVRG method that estimates the gradient of function, but considers the independence of two random variables $i_k$ and $j_k$, which are uniformly sampled from $\{1, 2, \cdots, n\}$ and $\{1, 2, \cdots, m\}$, that is $E[(\partial G_{j_k}(x))^{\top} F_{i_k}(y)] = (\partial G(x))^{\top} F(y)$. Furthermore, $G\left(\tilde{x}_v\right)$ is estimated from (2.5) rather than sums of $G_j$. Thus, the estimate gradient of $f(x)$ is

$$\begin{aligned}\nabla \hat{f}_k(x_k) = (\partial G_{j_k}(x_k))^{\top} \nabla F_{i_k}(\hat{G}_k)\\ -(\partial G_{j_k}(\tilde{x}_v))^{\top} \nabla F_{i_k}(G(\tilde{x}_v)) + \nabla f(\tilde{x}_v),\end{aligned} \tag{2.6}$$

where $\hat{G}_k$ is defined in (2.5), $\nabla f\left(\tilde{x}_v\right) = 1/n\sum_{j=1}^{n} \nabla G\left(\tilde{x}_v\right) F_j\left(G\left(\tilde{x}_v\right)\right)$.

For the non-convex composition problem, when the sampling times $A$ approximates to infinite, the expectation of estimate (2.6) of function $\nabla f(x)$ approximately equals to (2.2), that is

$$\mathbb{E}[\nabla \hat{f}_k(x_k)] \approx \mathbb{E}[(\partial G_{j_k}(x_k))^{\top} \nabla F_{i_k}(G(x_k))]. \tag{2.7}$$

This is because the infinite sample approximates to sums of function $G_i$, that is $\hat{G}_k \approx G\left(x_k\right)$. Our analysis also shows that SVRG for non-convex stochastic problems and non-convex composition problems have the same convergence rate. Moreover if using the estimated gradient $\nabla \tilde{f}\left(x_k\right)$ in (2.2) to compute the gradient information of composition problem, that is apply $G(x_k)$ instead of $\tilde{G}_k$, $m$ queries will be needed to compute $G(x_k)$, which leads to a question on how many sampling times $A$ is needed when the best query complexity is achieved.

## 3 Algorithm Design

In this section, we will design the risk-sensitive and reliable policy gradient-based algorithms. The objective function $J_\lambda(\theta)$ is described as follows:

$$J_\lambda(\theta) = \mathbb{E}_\pi[R(\theta)] - \lambda \mathrm{Var}[R(\theta)]. \tag{3.1}$$

We denote $J(s_t) = \mathbb{E}_\pi[R(\theta)|s_t]$, and $\mathrm{Var}(s_t) = \mathrm{Var}[R(\theta)|s_t]$. The gradient approach to solve $J_\lambda(\theta)$ is

$$\theta_{t+1} = \theta_t + \eta_t\left(\nabla J(\theta_t) - \lambda \nabla \mathrm{Var}(\theta_t)\right), \tag{3.2}$$

where $\eta_t$ is the stepsize. The sample-based estimation of $\nabla \mathrm{Var}(\theta)$ is computed as

$$\nabla \mathrm{Var}(\theta) = \nabla M(\theta) - 2J(\theta)\nabla J(\theta).$$

The computation of Eq. (3.2), therefore, involves the computation of the following three items, i.e. $\nabla J(\theta), \nabla M(\theta)$, and $J(\theta)\nabla J(\theta)$. The unbiased estimates of the two gradients, $\nabla J(s_t)$ and $\nabla M(s_t)$, can be obtained from a single trajectory via the likelihood ratio method as follows:

$$\nabla J(\theta) = \mathbb{E}[R_1^\tau \omega_\tau(\theta)], \quad \nabla M(\theta) = \mathbb{E}[(R_1^\tau)^2 \omega_\tau(\theta)],$$

where $R_1^\tau = \sum_{t=1}^{\tau} r_t$ is accumulated rewards, $\tau$ is the length of a trajectory, $t$ is the number of time steps in a trajectory and $\omega_\tau(\theta)$ is likelihood ratio derivative, which is computed as

$$\omega_\tau(\theta) = \sum_{t=1}^{\tau} \nabla \ln \pi_\theta(a_t|s_t). \tag{3.3}$$

The sample-based estimation of $J(s_t)\nabla J(s_t)$, however, is more complicated. If a generative model is available, i.e., for every state-action pair $(s, a)$, we can sample twice (or even multiple times) from the underlying Markov chain to obtain the successive state $s'$, and thus the sample-based estimation of the above three items are all available. However, if a generative model is not available, then the computation of $J(s_t)\nabla J(s_t)$ requires double sampling and can not be exactly estimated from a single trajectory, as also pointed out in (Tamar, Castro, and Mannor 2012) (The reason is similar to the fact that the variance of a random variable can not be estimated from a single trajectory). To this end, we are going to introduce the stochastic composite gradient (Lian, Wang, and Liu 2016) technique to address this problem.

### 3.1 Stochastic Composite Policy Gradient

In this section, we will use accelerated stochastic composite gradient descent method(Wang, Fang, and Liu 2017) to address this problem. We first rewrite the objective function Eq.(3.1) as follows:

$$\max_\theta \hat{J}_\lambda(\theta) = \left(J(\theta) - \lambda(M(\theta) - J(\theta)^2)\right), \tag{3.4}$$

where

$$J(\theta) = \sum_{t=1}^{\tau} r_t, M(\theta) = (\sum_{t=1}^{\tau} r_t)^2. \tag{3.5}$$

And next, we can rewrite it as the nested function form:

$$F_i(y) = \left(y_0 - \lambda\left(y_1 - y_0^2\right)\right)_i \tag{3.6}$$

$$G_j(\theta) = \left([R_1^\tau, (R_1^\tau)^2]^\top\right)_i \tag{3.7}$$

$y \in \mathbb{R}^2$, and $y_0$, $y_1$ denote the first and second entry of $y$, respectively. Here the outer function is known, its gradient can be calculated, and we can get the gradient of the inner function by likelihood ratio derivative:

$$\nabla F_i(y) = \left([1 + 2\lambda y_0, -\lambda]^\top\right)_i \tag{3.8}$$

$$\nabla G_j(\theta) = \left([R_1^\tau \omega_\tau(\theta), (R_1^\tau)^2 \omega_\tau(\theta)]^\top\right)_j \tag{3.9}$$

Algorithm 1 describes the stochastic composition policy gradient method.

### 3.2 R$^2$ Policy Gradient

It is interesting to reduce the variance of Algorithm 1 even further. The original mean-variance objective function is reformulated in the finite-sum form:

$$\max_\theta \hat{J}_\lambda(\theta) = \left(J(\theta) - \lambda(M(\theta) - J(\theta)^2)\right), \tag{3.10}$$

where

$$J(\theta) = \sum_{t=1}^{\tau} r_t, M(\theta) = (\sum_{t=1}^{\tau} r_t)^2. \tag{3.11}$$

which can be reformulated into the composite finite-sum structure as follows:

$$F_i(y) = (y_1 - \lambda\left(y_1 - y_0^2\right))_i$$

$$G_j(\theta) = ([R_1^\tau, (R_1^\tau)^2]^\top)_j,$$

**Algorithm 1** Stochastic Composition Optimization (Wang, Fang, and Liu 2017)

```
Require: $K$, $\beta_k$,$\eta_k$ (learning rate)
Ensure: Initialize $\theta_0$, $y_0$.
  for $k = 1, 2, 3, \cdots, K$ do
    Uniformly randomly pick $i_k$ and $j_k$ from $\{1, ..., n\}$ and $\{1, ..., m\}$
    Compute the value of $G$ at $\theta_k$ to obtain $G_{j_k}(\theta_k)$ and $\nabla G_{j_k}(\theta_k)$.
        $y_{k+1} = (1 - \beta_k)y_k + \beta_k G_{j_k}(\theta_k)$
    Compute $\nabla F$ at $y_{k+1}$ to obtain $\nabla F_{i_k}(y_{k+1})$.
        $\theta_{k+1} = \theta_k + \eta_k(\nabla G_{j_k}(\theta_k))^T \nabla F_{i_k}(y_{k+1})$,
  end for
```

where $(\cdot)_i$ (resp. $(\cdot)_j$) denotes the $i$-th (resp. $j$-th) trajectory. $y \in \mathbb{R}^2$, and $y_0$, $y_1$ denote the first and second entry of $y$, respectively. Here the outer function is known, its gradient can be calculated, and we can get the gradient of the inner function by likelihood ratio derivative:

$$\nabla F_i(y) = \Big([1 + 2\lambda y_0, -\lambda]^\top\Big)_i$$

$$\nabla G_j(\theta) = \Big([R_1^\tau \omega_\tau(\theta), (R_1^\tau)^2 \omega_\tau(\theta)]^\top\Big)_j$$

R$^2$PG algorithm is introduced in Algorithm 2.

## 4 Experimental Study

### 4.1 Portfolio Management

We conduct empirical studies for the proposed algorithm by comparing to the state-of-the-art risk sensitive policy gradient algorithm by Tamar et al. (Tamar, Castro, and Mannor 2012). We consider a portfolio domain in (Tamar, Castro, and Mannor 2012) that is composed of two types of assets. A liquid asset has a fixed interest rate $r_l$ and can be sold at any time step $t = 1, \cdots, T$. A non-liquid asset has a time related interest rate $r_{nl}(t)$, and can be sold only after a maturity period of $N$ steps. Besides, the non-liquid asset surfers a default risk with a probability $p_{risk}$. In our model, the investor may change his portfolio by investing a fixed fraction $\alpha$ of his total available cash in a non-liquid asset at each time step. We assume that at each $t$ the interest rate $r_{nl}(t)$ high-low takes one of two values $r_{nl}^{low}$ or $r_{nl}^{high}$, and the transitions between these values occur stochastically with switching probability $p_{switch}$. The state of the model at each time step is represented by a vector $x(t) \in R^{N+2}$ ,where $x_1 \in [0, 1]$ is the fraction of the investment in liquid assets, $x_2, \cdots, x_{N+1} \in [0, 1]$ is the fraction in non-liquid assets with time to maturity of $1, \cdots, N$ time steps, respectively, and $x_{N+2}(t) = r_{nl}(t) - \mathbb{E}[r_{nl}(t)]$. We assume that all investments are in liquid assets at time $t = 0$, the startup cash is 100. Our reward is just logarithm of return rate. The

**Algorithm 2** Robust and Reliable Policy Gradient (R$^2$PG)

```
Require: $K$(Inner iteration), $V$ (Outer iteration), $\eta$ (learning rate), and $\tilde{\theta}_1$.
  for $v = 1, 2, \cdots, V$ do
        $G(\tilde{\theta}_v) = \frac{1}{m}\sum_{j=1}^{m} G_j(\tilde{\theta}_v)$        (3.12)
        $\nabla f(\tilde{\theta}_v) = \frac{1}{n}\sum_{i=1}^{n} F_i(G(\tilde{\theta}_v))$        (3.13)
        $\theta_1 = \tilde{\theta}_v$        (3.14)
    for $k = 1, 2, \cdots, K$ do
      Sample from $\{1, ..., m\}$ for A times to form mini-batch multiset $\mathcal{A}_k$
      Estimate $\hat{G}_k$ from (2.5)
      Uniformly randomly pick $i_k$ and $j_k$ from $\{1, ..., n\}$ and $\{1, ..., m\}$
      Estimate $\nabla \hat{f}_k(\theta_k)$ from (2.6)
            $\theta_{k+1} = \theta_k + \eta \nabla \hat{f}_k(\theta_k)$
    end for
    $\tilde{\theta}_{v+1}$ is randomly chosen from $\{\theta_k\}$, $k \in \{1, \cdots, K\}$
  end for
```

binary action at each step is determined by a softmax policy as $\pi(a|s,\theta) = \frac{e^{\phi(s,a)^\top \theta}}{\sum e^{\phi(s,\cdot)^\top \theta}}$. We conduct empirical studies for the proposed two algorithms by comparing them to Tamar's algorithms. Figure 1 shows the distribution of the accumulated reward. As anticipated, the R$^2$ PG method got the low-variance and better mean than Tamar's method and stochastic composite policy gradient method due to applying the variance reduction technique.

## 4.2 American-style Option

We also apply $R^2$PG algorithm on an American-style Option(call & put) domain in (Tamar, Mannor, and Xu 2014). American-style option is a contract which gives the buyer (the owner or holder of the option) the right, but not the obligation, to buy or sell an underlying asset at a specific strike price $K$ at before a specified date, or before some maturity time $T$. We represent the state $x_t$ as the price of the asset at time $t \leq T$, the reward of executing a put option at that time is $g_{put}(x_t)$, where $g_{put}(x) = max(0, K - x)$, whereas for a call option we have $g_{call}(x) = max(0, x - K)$. This American-style may be formulated as an Random MDP as follows. The state at $t$ time is $\{x_t, t\}$. The action is binary, where 1 stands for executing the option and 0 for continuing to hold it. Once an option is executed, or $t = T$, a transition to terminal state occurs and the reward will be given. Otherwise, the state transits to next state $\{x_{t+1}, t+1\}$ where $x_{t+1}$ id determined by a stochastic kernel $\hat{P}(x'|x, t)$. The reward for executing $a = 1$ at state $x$ is $g(x)$ and 0 if $a = 0$ or $t = T$.

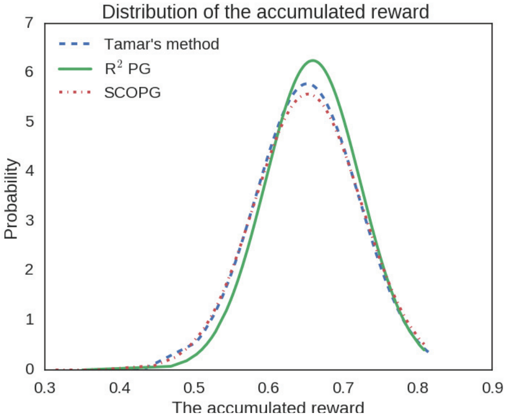

Figure 1: Distribution of the accumulated reward

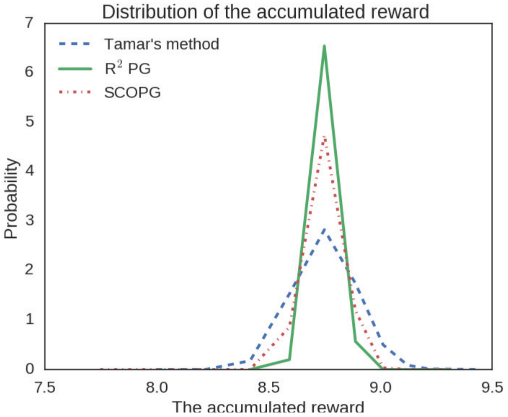

Figure 2: Distribution of the accumulated reward

We consider a mixed investment $g(x) = g_{put}(x) + g_{call}(x)$. Our price fluctuation model $M$ follows a Bernoulli distribution, $x_{t+1} = \begin{cases} f_u x_t, \text{w.p.} & p \\ f_d x_t, \text{w.p.} & 1-p \end{cases}$ ,where the up and down factors, $f_u$ and $f_d$ are constant. We assume that $K$ is equal to the initial price $x_0$ at time $t = 0$. The binary action at each step is determined by a softmax policy as $\pi(a|s,\theta) = \frac{e^{\phi(s,a)^\top \theta}}{\sum e^{\phi(s,\cdot)^\top \theta}}$. We conduct empirical studies for the proposed two algorithms by comparing them to Tamar's algorithms. Figure 2 shows the distribution of the accumulated reward. As anticipated, the R$^2$ PG method got the low-variance and better mean than Tamar's method and stochastic composite policy gradient method due to applying the variance reduction technique.

## 5 Conclusions

There are many interesting future directions along this research topic. Besides stochastic policy gradient, deterministic policy gradient (Silver et al. 2014) has shown great potential in large discrete action space. Secondly, it is interesting to explore other variance techniques based on the finite-sum structure. It should be noted that SVRG/SAGA is not the only solution technique for such finite-sum structured problems. There might be other methods which have not been fully explored, which are worthwhile trying in the policy search framework. Another interesting future direction is to explore other interesting variance reduction techniques in policy gradient approaches other than utilizing the finite-sum problem structure. Previous algorithms rely on control variate methods such as the baseline method and actor-critic method, and in this paper variance reduction based on finite-sum structure is explored. It is interesting to see if other stochastic variance reduction technique can be used.

## References

Chow, Y., and Ghavamzadeh, M. 2014. Algorithms for cvar optimization in mdps. In *Advances in Neural Information Processing Systems*, 3509–3517.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, 1646–1654.

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

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 315–323.

Lian, X.; Wang, M.; and Liu, J. 2016. Finite-sum composition optimization via variance reduced gradient descent. *arXiv preprint arXiv:1610.04674*.

Palaniappan, B., and Bach, F. 2016. Stochastic variance reduction methods for saddle-point problems. In *NIPS*. 1416–1424.

Peters, J., and Schaal, S. 2006. Policy gradient methods for robotics. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2219–2225. IEEE.

Prashanth, L. A., and Ghavamzadeh, M. 2013. Actor-critic algorithms for risk-sensitive mdps. In *Advances in Neural Information Processing Systems*, 252–260.

Reddi, S.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. In *ICML*.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *ICML*, 387–395.

Tamar, A.; Castro, D. D.; and Mannor, S. 2012. Policy gradients with variance related risk criteria. In *ICML*, 935–942.

Tamar, A.; Mannor, S.; and Xu, H. 2014. Scaling up robust mdps using function approximation. In *International Conference on Machine Learning*, 181–189.

Wang, M.; Fang, E. X.; and Liu, H. 2017. Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions. *Mathematical Programming* 161(1-2):419–449.

Wang, M.; Liu, J.; and Fang, E. 2016. Accelerating stochastic composition optimization. In *Advances in Neural Information Processing Systems*, 1714–1722.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Zhang, H.; Reddi, S.; and Sra, S. 2016. Riemannian SVRG: Fast Stochastic Optimization on Riemannian Manifolds. In *NIPS*. 4592–4600.

Zhu, Z., and Hazan, E. 2016. Variance reduction for faster non-convex optimization. *arXiv preprint arXiv:1603.05643*.

Zhu, Z., and Yuan, Y. 2016. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In *ICML*.