Noname manuscript No.
(will be inserted by the editor)

Variance-Constrained Actor-Critic Algorithms for Discounted and Average Reward MDPs

Prashanth L.A.[†] · **Mohammad Ghavamzadeh**[♯]

Abstract In many sequential decision-making problems we may want to manage risk by minimizing some measure of variability in rewards in addition to maximizing a standard criterion. Variance related risk measures are among the most common risk-sensitive criteria in finance and operations research. However, optimizing many such criteria is known to be a hard problem. In this paper, we consider both discounted and average reward Markov decision processes. For each formulation, we first define a measure of variability for a policy, which in turn gives us a set of risk-sensitive criteria to optimize. For each of these criteria, we derive a formula for computing its gradient. We then devise actor-critic algorithms that operate on three timescales - a TD critic on the fastest timescale, a policy gradient (actor) on the intermediate timescale, and a dual ascent for Lagrange multipliers on the slowest timescale. In the discounted setting, we point out the difficulty in estimating the gradient of the variance of the return and incorporate simultaneous perturbation approaches to alleviate this. The average setting, on the other hand, allows for an actor update using compatible features to estimate the gradient of the variance. We establish the convergence of our algorithms to locally risk-sensitive optimal policies. Finally, we demonstrate the usefulness of our algorithms in a traffic signal control application.

Keywords Markov decision process (MDP) · reinforcement learning (RL) · risk sensitive RL · actor-critic algorithms · multi-time-scale stochastic approximation · simultaneous perturbation stochastic approximation (SPSA) · smoothed functional (SF).

1 Introduction

The usual optimization criteria for an infinite horizon Markov decision process (MDP) are the *expected sum of discounted rewards* and the *average reward* [50, 5]. Many algorithms have been developed to maximize these criteria both when the model of the system is known (planning) and unknown (learning) [7, 64]. These algorithms can be categorized to **value function-based** methods that are mainly based on the two celebrated dynamic programming algorithms *value iteration* and *policy iteration*; and **policy gradient** methods that are

[†] INRIA Lille - Team SequeL, E-mail: prashanth.la@inria.fr

[♯] Adobe Research & INRIA Lille - Team SequeL *(currently at Adobe Research, on leave of absence from INRIA)*, E-mail: mohammad.ghavamzadeh@inria.fr

based on updating the policy parameters in the direction of the gradient of a performance measure, i.e., the value function of the initial state or the average reward. Policy gradient methods estimate the gradient of the performance measure either without using an explicit representation of the value function (e.g., [73, 39, 4]) or using such a representation in which case they are referred to as *actor-critic* algorithms (e.g., [65, 33, 44, 13, 14]). Using an explicit representation for value function (e.g., linear function approximation) by actor-critic algorithms reduces the variance of the gradient estimate with the cost of adding it a bias.

Actor-critic methods were among the earliest to be investigated in RL [2, 62]. They comprise a family of reinforcement learning (RL) methods that maintain two distinct algorithmic components: An *Actor*, whose role is to maintain and update an action-selection policy; and a *Critic*, whose role is to estimate the value function associated with the actor's policy. Thus, the critic addresses a problem of *prediction*, whereas the actor is concerned with *control*. A common practice is to update the policy parameters using stochastic gradient ascent, and to estimate the value-function using some form of temporal difference (TD) learning [63].

However in many applications, we may prefer to minimize some measure of *risk* as well as maximizing a usual optimization criterion. In such cases, we would like to use a criterion that incorporates a penalty for the *variability* induced by a given policy. This variability can be due to two types of uncertainties: **(i)** uncertainties in the model parameters, which is the topic of *robust* MDPs (e.g., [43, 25, 74]), and **(ii)** the inherent uncertainty related to the stochastic nature of the system, which is the topic of *risk-sensitive* MDPs (e.g., [31, 57, 28]).

In risk-sensitive sequential decision-making, the objective is to maximize a risk-sensitive criterion such as the expected exponential utility [31], a variance related measure [57, 28], the percentile performance [29], or conditional value-at-risk (CVaR) [52, 55]. Unfortunately, when we include a measure of risk in our optimality criteria, the corresponding optimal policy is usually no longer Markovian stationary (e.g., [28]) and/or computing it is not tractable (e.g., [28, 37]). In particular, **(i)** In [57], the author analyzed variance constraints in the context of a discounted reward MDP and showed the existence of a Bellman equation for the variance of the return. However, it was established there that the operator underlying the aforementioned Bellman equation is *not necessarily* monotone. The latter is a crucial requirement for employing popular dynamic programming procedures for solving MDPs. **(ii)** In [38], the authors provide hardness results for variance constrained MDPs and in particular show that finding a globally mean-variance optimal policy in a discounted MDP is NP-hard, even when the underlying transition dynamics are known. **(iii)** In [28], the authors established hardness results for average reward MDP, with a variance constraint that differs significantly from its counterpart in the discounted setting. Nevertheless, the variance constraint is well motivated considering the objective is to optimize a long-run average reward. However, the mathematical difficulties in finding a globally mean variance optimal policy remains, even with this altered variance constraint.

Although risk-sensitive sequential decision-making has a long history in operations research and finance, it has only recently grabbed attention in the machine learning community. Most of the work on this topic (including those mentioned above) has been in the context of MDPs (when the model of the system is known) and much less work has been done within the reinforcement learning (RL) framework (when the model is unknown and all the information about the system is obtained from the samples resulted from the agent's interaction with the environment). In risk-sensitive RL, we can mention the work by Borkar [18, 19, 22] and Basu et al. [3] who considered the expected exponential utility, the one by Mihatsch and Neuneier [41] that formulated a new risk-sensitive control framework based on transforming the temporal difference errors that occur during learning, and the one by Tamar et al. [68] on several variance related measures. Tamar et al. [68] study stochastic shortest path problems, and in this context, propose a policy gradient algorithm (and in a more recent work [67] an actor-critic algorithm) for maximizing several risk-sensitive criteria that involve both the expectation and variance of the *return* random variable (defined as the sum of the rewards that the agent obtains in an episode).

In this paper,[1] we develop actor-critic algorithms for optimizing variance-related risk measures in both discounted and average reward MDPs. In the following, we first summarize our contributions in the discounted reward setting and follow it with those in average reward setting.

Discounted reward setting. Here we define the measure of variability as the *variance of the return* (similar to [68]). We formulate the following constrained optimization problem with the aim of maximizing the mean of the return subject to its variance being bounded from above: For a given $\alpha > 0$,

$$\max_{\theta} V^{\theta}(x^0) \qquad \text{subject to} \qquad \Lambda^{\theta}(x^0) \leq \alpha.$$

In the above, $V^{\theta}(x^0)$ is the mean of the return, starting in state x^0 for a policy identified by its parameter θ, while $\Lambda^{\theta}(x^0)$ is the variance of the return (see Section 3 for precise definitions). A standard approach to solve the above problem is to employ the Lagrangian relaxation procedure [6] and solve the following unconstrained problem:

$$\max_{\lambda} \min_{\theta} \Big(L(\theta, \lambda) \overset{\triangle}{=} -V^{\theta}(x^0) + \lambda\big(\Lambda^{\theta}(x^0) - \alpha\big) \Big),$$

where λ is the Lagrange multiplier. For solving the above problem, it is required to derive a formula for the gradient of the Lagrangian $L(\theta, \lambda)$, both w.r.t. θ and λ. While the gradient w.r.t. λ is particularly simple since it is the constraint value, the other gradient, i.e., w.r.t. θ is complicated. We derive this formula in Lemma 1 and show that $\nabla_{\theta} L(\theta, \lambda)$ requires the gradient of the value function at every state of the MDP (see the discussion in Sections 3 and 4).

Note that we operate in a *simulation optimization* setting, i.e., we have access to reward samples from the underlying MDP. Thus, it is required to estimate the mean and variance of the return (we use a TD-critic for this purpose) and then use these estimates to compute gradient of the Lagrangian. The latter is used then used to descend in the policy parameter. We estimate the gradient of the Lagrangian using two simultaneous perturbation methods: *simultaneous perturbation stochastic approximation* (SPSA) [58] and *smoothed functional* (SF) [32], resulting in two separate discounted reward actor-critic algorithms. In addition, we also propose second-order algorithms with a Newton step, using both SPSA and SF.

Simultaneous perturbation methods have been popular in the field of stochastic optimization and the reader is referred to [17] for a textbook introduction. First introduced in [58], the idea of SPSA is to perturb each coordinate of a parameter vector uniformly using a Rademacher random variable, in the quest for finding the minimum of a function that is only observable via simulation. Traditional gradient schemes require $2\kappa_1$ evaluations of the function, where κ_1 is the parameter dimension. On the other hand, SPSA requires only two evaluations irrespective of the parameter dimension and hence is an efficient scheme, especially useful in high-dimensional settings. While a one-simulation variant of SPSA was proposed in [59], the original two-simulation SPSA algorithm is preferred as it is more efficient and also seen to work better than its one-simulation variant. Later enhancements to

[1] This paper is an extension of an earlier work by the authors [48] and includes novel second order methods in the discounted setting, detailed proofs of all proposed algorithms, and additional experimental results.

the original SPSA scheme include using deterministic perturbation using certain Hadamard matrices [12] and second-order methods that estimate Hessian using SPSA [60, 8]. The SF schemes are another class of simultaneous perturbation methods, which again perturb each coordinate of the parameter vector uniformly. However, unlike SPSA, Gaussian random variables are used here for the perturbation. Originally proposed in [32], the SF schemes have been studied and enhanced in later works such as [61, 9]. Further, [16] proposes both SPSA and SF like schemes for constrained optimization.

Average reward setting. Here we first define the measure of variability as the *long-run variance* of a policy as follows:

$$\Lambda(\theta) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{n=0}^{T-1} \big(R_n - \rho(\mu)\big)^2 \middle| \theta\right],$$

where $\rho(\theta)$ is the average reward under policy identified by its parameter θ (see Section 5 for precise definitions). The aim here is to solve the following constrained optimization problem:

$$\max_{\theta} \rho(\theta) \qquad \text{subject to} \qquad \Lambda(\theta) \leq \alpha.$$

As in the discounted setting. we derive an expression for the gradient of the Lagrangian (see Lemma 3). Unlike the discounted setting, we do not require sophisticated simulation optimizations schemes, as the gradient expressions in Lemma 3 suggest a simpler alternative that employs *compatible features* [65, 44]. Compatible features for linearly approximating the action-value function of policy θ are of the form $\nabla \log \mu(a|x)$. These features are well-defined if the policy is differentiable w.r.t. its parameters θ. Sutton et al [65] showed the advantages of using these features in approximating the action-value function in actor-critic algorithms. In [14], the authors use compatible features to develop actor-critic algorithms for a risk-neutral setting. We extend this to variance-constrained setting and establish that the square value function itself serves as a good baseline level when calculating the gradient of the average square reward (see the discussion surrounding Lemma 4). This facilitates the usage of *compatible features* for obtaining unbiased estimates of both average reward as well as square reward. We then develop an actor-critic algorithm that employ these *compatible features* in order to descend in the policy parameter θ and also identify the bias that arises due to function approximation (see Lemma 5).

Proof of convergence. Using the ordinary differential equations (ODE) approach, we establish the asymptotic convergence of our algorithms to locally risk-sensitive optimal policies and in the light of hardness results from [38], this is the best one can hope to achieve. Our algorithms employ multi-timescale stochastic approximation, in both settings. The convergence proof proceeds by analysing each timescale separately. In essence, the iterates on a faster timescale view those on a slower timescale as quasi-static, while the slower timescale iterate views that on a faster timescale as equilibrated. Using this principle, we show that TD critic (on the fastest timescale in all the algorithms) converge to fixed points of the Bellman operator, for any fixed policy θ and Lagrange multiplier λ. Next, for any given λ, the policy update tracks in the asymptotic limit and converges to the equilibria of the corresponding ODE. Finally, λ updates on slowest timescale converge and the overall convergence is to a local saddle point of the Lagrangian. Moreover, the limiting point is feasible for the constrained optimization problem mentioned above, i.e., the policy obtained upon convergence satisfies the constraint that the variance is upper-bounded by α.

Simulation experiments. We demonstrate the usefulness of our discounted and average reward risk-sensitive actor-critic algorithms in a traffic signal control application. On this high-dimensional system with state space $\approx 10^{32}$, the objective in our formulation is to minimize the total number of vehicles in the system, which indirectly minimizes the delay experienced by the system. The motivation behind using a risk-sensitive control strategy is to reduce the variations in the delay experienced by road users. From the results, we observe that the risk-sensitive algorithms proposed in this paper result in a long-term (discounted or average) cost that is higher than their risk-neutral variants. However, from the empirical variance of the cost (both discounted as well as average) perspective, the risk-sensitive algorithms outperform their risk-neutral variants. Moreover, the experiments in the discounted setting also show that our SPSA based actor-critic scheme outperforms the policy gradient algorithm proposed in [68], both from a mean-variance as well as gradient estimation standpoints. This observation justifies using the actor-critic approach for solving risk-sensitive MDPs, as it reduces the variance of the gradient estimated by the policy gradient approach with the cost of introducing a bias induced by the value function representation.

Remark 1 It is important to note that both our discounted and average reward algorithms can be easily extended to other variance related risk criteria such as the Sharpe ratio, which is popular in financial decision-making [54] (see Remarks 3 and 7 for more details).

Remark 2 Another important point is that the *expected exponential utility* risk measure can be also considered as an approximation of the mean-variance tradeoff due to the following Taylor expansion (see e.g., Eq. 11 in [41])

$$-\frac{1}{\beta}\log\mathbb{E}[e^{-\beta X}] = \mathbb{E}[X] - \frac{\beta}{2}\text{Var}[X] + O(\beta^2),$$

and we know that it is much easier to design actor-critic or other reinforcement learning algorithms [18, 19, 3, 22] for this risk measure than those that will be presented in this paper. However, this formulation is limited in the sense that it requires knowing the ideal tradeoff between the mean and variance, since it takes β as an input. On the other hand, the mean-variance formulations considered in this paper are more general because
(i) we optimize for the Lagrange multiplier λ, which plays a similar role to β, as a tradeoff between the mean and variance, and
(ii) it is usually more natural to know an upper-bound on the variance (as in the mean-variance formulations considered in this paper) than knowing the ideal tradeoff between the mean and variance (as considered in the expected exponential utility formulation).
Despite all these, we should not consider these formulations as replacement for each other or try to find a formulation that is the best for all problems, but instead should consider them as different formulations that each might be the right fit for a specific problem.

Closely related works. In comparison to [68] and [67], which are the most closely related contributions, we would like to point out the following:
(i) The authors develop policy gradient and actor-critic methods for stochastic shortest path problems in [68] and [67], respectively. On the other hand, we devise actor-critic algorithms for both discounted and average reward MDP settings; and
(ii) More importantly, we note the difficulty in the discounted formulation that requires to estimate the gradient of the value function at every state of the MDP and also sample from two different distributions. This precludes us from using *compatible features* - a method that has been employed successfully in actor-critic algorithms in a risk-neutral setting (cf. [14])

as well as more recently in [67] for a risk-sensitive stochastic shortest path setting. We alleviate the above mentioned problems for the discounted setting by employing simultaneous perturbation based schemes for estimating the gradient in the first order methods and Hessian in the second order methods, that we propose.
(iii) Unlike [68, 67] who consider a fixed λ in their constrained formulations, we perform dual ascent using sample variance constrants and optimize the Lagrange multiplier λ. In rigorous terms, λ_n in our algorithms is shown to converge to a local maxima of $\nabla_\lambda L(\theta^\lambda, \lambda)$ (here θ^λ is the limit of the θ recursion for a given value of λ) and the limit λ^* is such that the variance constraint is satisfied for the corresponding policy θ^{λ^*}.

Organization of the paper. The rest of the paper is organized as follows: In Section 2, we describe the RL setting. In Section 3, we describe the risk-sensitive MDP in the discounted setting and propose actor-critic algorithms for this setting in Section 4. In Section 5, we present the risk measure for the average setting and propose an actor-critic algorithm that optimizes this risk measure in Section 6. In Sections 7–8, we present the convergence proofs for the algorithms in discounted and average reward settings, respectively. In Section 9, we describe the experimental setup and present the results in both average and discounted cost settings. Finally, in Section 10, we provide the concluding remarks and outline a few future research directions.

2 Preliminaries

We consider sequential decision-making tasks that can be formulated as a reinforcement learning (RL) problem. In RL, an agent interacts with a dynamic, stochastic, and incompletely known environment, with the goal of optimizing some measure of its *long-term* performance. This interaction is often modeled as a Markov decision process (MDP). A MDP is a tuple $(\mathcal{X}, \mathcal{A}, R, P, x^0)$ where $\mathcal{X}$ and $\mathcal{A}$ are the state and action spaces; $R(x,a), x \in \mathcal{X}, a \in \mathcal{A}$ is the reward random variable whose expectation is denoted by $r(x,a) = \mathbb{E}\big[R(x,a)\big]$; $P(\cdot|x,a)$ is the transition probability distribution; and $x^0 \in \mathcal{X}$ is the initial state[2]. We assume that both state and action spaces are finite.

The rule according to which the agent acts in its environment (selects action at each state) is called a *policy*. A Markovian stationary policy $\mu(\cdot|x)$ is a probability distribution over actions, conditioned on the current state x. The goal in a RL problem is to find a policy that optimizes the long-term performance measure of interest, e.g., maximizes the *expected discounted sum of rewards* or the *average reward*.

In policy gradient and actor-critic methods, we define a class of parameterized stochastic policies $\big\{\mu(\cdot|x;\theta), x \in \mathcal{X}, \theta \in \Theta \subseteq \mathbb{R}^{\kappa_1}\big\}$, estimate the gradient of the performance measure w.r.t. the policy parameters θ from the observed system trajectories, and then improve the policy by adjusting its parameters in the direction of the gradient. Here Θ denotes a compact and convex subset of $\mathbb{R}^{\kappa_1}$. Our algorithms projects the iterates onto Θ, which ensures stability - a crucial requirement necessary for establishing convergence. Since in this setting a policy μ is represented by its κ_1-dimensional parameter vector θ, policy dependent functions can be written as a function of θ in place of μ. So, we use μ and θ interchangeably in the paper.

We make the following assumptions on the policy, parameterized by θ:

[2] Our algorithms can be easily extended to a setting where the initial state is determined by a distribution.

(A1) *For any state-action pair* $(x,a) \in \mathcal{X} \times \mathcal{A}$*, the policy* $\mu(a|x;\theta)$ *is continuously differentiable in the parameter* θ*.*
(A2) *The Markov chain induced by any policy* θ *is irreducible.*

The above assumptions are standard requirements in policy gradient and actor-critic methods.

Finally, we denote by $d^\mu(x)$ and $\pi^\mu(x,a) = d^\mu(x)\mu(a|x)$, the stationary distribution of state x and state-action pair (x,a) under policy μ, respectively. The stationary distributions can be seen to exist because we consider a finite state-action space setting and irreducibility here implies positive recurrence. Similarly in the discounted formulation, we define the γ-discounted visiting distribution of state x and state-action pair (x,a) under policy μ as $d^\mu_\gamma(x|x^0) = (1-\gamma)\sum_{n=0}^\infty \gamma^n \Pr(x_n = x|x_0 = x^0;\mu)$ and $\pi^\mu_\gamma(x,a|x^0) = d^\mu_\gamma(x|x^0)\mu(a|x)$.

3 Discounted Reward Setting

For a given policy μ, we define the return of a state x (state-action pair (x,a)) as the sum of discounted rewards encountered by the agent when it starts at state x (state-action pair (x,a)) and then follows policy μ, i.e.,

$$D^\mu(x) = \sum_{n=0}^\infty \gamma^n R(x_n,a_n) \mid x_0 = x,\ \mu,$$
$$D^\mu(x,a) = \sum_{n=0}^\infty \gamma^n R(x_n,a_n) \mid x_0 = x,\ a_0 = a,\ \mu.$$

The expected value of these two random variables are the value and action-value functions of policy μ, i.e.,

$$V^\mu(x) = \mathbb{E}\big[D^\mu(x)\big] \qquad \text{and} \qquad Q^\mu(x,a) = \mathbb{E}\big[D^\mu(x,a)\big].$$

The goal in the standard (risk-neutral) discounted reward formulation is to find an optimal policy $\mu^* = \arg\max_\mu V^\mu(x^0)$, where x^0 is the initial state of the system.

The most common measure of the *variability* in the stream of rewards is the *variance of the return*, defined by

$$\Lambda^\mu(x) \triangleq \mathbb{E}\big[D^\mu(x)^2\big] - V^\mu(x)^2 = U^\mu(x) - V^\mu(x)^2. \tag{1}$$

The above measure was first introduced by Sobel [57]. Note that

$$U^\mu(x) \triangleq \mathbb{E}\left[D^\mu(x)^2\right]$$

is the *square reward value function* of state x under policy μ. On similar lines, we define the *square reward action-value function* of state-action pair (x,a) under policy μ as

$$W^\mu(x,a) \triangleq \mathbb{E}\left[D^\mu(x,a)^2\right].$$

From the Bellman equation of $\Lambda^\mu(x)$, proposed by Sobel [57], it is straightforward to derive the following Bellman equations for $U^\mu(x)$ and $W^\mu(x,a)$:

$$\begin{aligned} U^\mu(x) &= \sum_a \mu(a|x) r(x,a)^2 + \gamma^2 \sum_{a,x'} \mu(a|x) P(x'|x,a) U^\mu(x') \\ &\quad + 2\gamma \sum_{a,x'} \mu(a|x) P(x'|x,a) r(x,a) V^\mu(x'), \qquad (2) \\ W^\mu(x,a) &= r(x,a)^2 + \gamma^2 \sum_{x'} P(x'|x,a) U^\mu(x') + 2\gamma r(x,a) \sum_{x'} P(x'|x,a) V^\mu(x'). \end{aligned}$$

Although Λ^μ of (1) satisfies a Bellman equation, unfortunately, it lacks the monotonicity property of dynamic programming (DP), and thus, it is not clear how the related risk measures can be optimized by standard DP algorithms [57]. Policy gradient and actor-critic algorithms are good candidates to deal with this risk measure.

We consider the following risk-sensitive measure for discounted MDPs: For a given $\alpha > 0$,

$$\max_\theta V^\theta(x^0) \qquad \text{subject to} \qquad \Lambda^\theta(x^0) \le \alpha. \qquad (3)$$

Assuming that there is at least one policy (in the class of parameterized policies that we consider) that satisfies the variance constraint above, it can be inferred from Theorem 3.8 of [1] that there exists an optimal policy that uses at most one randomization.

It is important to note that the algorithms proposed in this paper can be used for any risk-sensitive measure that is based on the variance of the return such as

1. $\min_\theta \Lambda^\theta(x^0) \qquad \text{subject to} \qquad V^\theta(x^0) \ge \alpha$,
2. $\max_\theta V^\theta(x^0) - \alpha\sqrt{\Lambda^\theta(x^0)}$,
3. Maximizing the Sharpe Ratio, i.e., $\max_\theta V^\theta(x^0)/\sqrt{\Lambda^\theta(x^0)}$. Sharpe Ratio (SR) is a popular risk measure in financial decision-making [54]. Section 3 presents extensions of our proposed discounted reward algorithms to optimize the Sharpe ration.

To solve (3), we employ the Lagrangian relaxation procedure [6] to convert it to the following unconstrained problem:

$$\max_\lambda \min_\theta \Big(L(\theta,\lambda) \overset{\triangle}{=} -V^\theta(x^0) + \lambda\big(\Lambda^\theta(x^0) - \alpha\big)\Big), \qquad (4)$$

where λ is the Lagrange multiplier. The goal here is to find the saddle point of $L(\theta,\lambda)$, i.e., a point (θ^*,λ^*) that satisfies

$$L(\theta,\lambda^*) \ge L(\theta^*,\lambda^*) \ge L(\theta^*,\lambda), \forall\theta \in \Theta, \forall\lambda > 0.$$

For a standard convex optimization problem where the objective $L(\theta,\lambda)$ is convex in θ and concave in λ, one can ensure the existence of a unique saddle point under mild regularity conditions (cf. [56]). Further, convergence to this point can be achieved by descending in θ and ascending in λ using $\nabla_\theta L(\theta,\lambda)$ and $\nabla_\lambda L(\theta,\lambda)$, respectively.

However, in our setting, the Lagrangian $L(\theta,\lambda)$ is not necessarily convex in θ, which implies there may not be an unique saddle point. The problem is further complicated by the fact that we operate in a *simulation optimization* setting, i.e., only sample estimates of the Lagrangian are obtained. Hence, performing primal descent and dual ascent, one can only get to a local saddle point, i.e., a tuple (θ^*,λ^*) which is a local minima w.r.t. θ and local maxima w.r.t λ of the Lagrangian. As an aside, global mean-variance optimization of MDPs have been shown to be NP-hard in [38] and the best one can hope is to find a approximately optimal policy.

In our setting, the necessary gradients of the Lagrangian are as follows:

$$\nabla_\theta L(\theta,\lambda) = -\nabla_\theta V^\theta(x^0) + \lambda \nabla_\theta \Lambda^\theta(x^0) \qquad \text{and} \qquad \nabla_\lambda L(\theta,\lambda) = \Lambda^\theta(x^0) - \alpha.$$

Since $\nabla_\theta \Lambda^\theta(x^0) = \nabla_\theta U^\theta(x^0) - 2V^\theta(x^0)\nabla_\theta V^\theta(x^0)$, in order to compute $\nabla_\theta \Lambda^\theta(x^0)$ it would be enough to calculate $\nabla_\theta V^\theta(x^0)$ and $\nabla_\theta U^\theta(x^0)$. Using the above definitions, we are now ready to derive the expressions for the gradient of $V^\theta(x^0)$ and $U^\theta(x^0)$, which in turn constitute the main ingredients in calculating $\nabla_\theta L(\theta,\lambda)$[3].

Lemma 1 *Under (A1) and (A2), we have*

$$\begin{aligned}
(1-\gamma)\nabla V^\theta(x^0) &= \sum_{x,a} \pi_\gamma^\theta(x,a|x^0)\nabla \log \mu(a|x;\theta) Q^\theta(x,a), \\
(1-\gamma^2)\nabla U^\theta(x^0) &= \sum_{x,a} \widetilde{\pi}_\gamma^\theta(x,a|x^0)\nabla \log \mu(a|x;\theta) W^\theta(x,a) \\
&\quad + 2\gamma \sum_{x,a,x'} \widetilde{\pi}_\gamma^\theta(x,a|x^0) P(x'|x,a) r(x,a) \nabla V^\theta(x'),
\end{aligned}$$

where $\widetilde{d}_\gamma^\theta(x|x^0)$ and $\widetilde{\pi}_\gamma^\theta(x,a|x^0)$ are the γ^2-discounted visiting distributions of state x and state-action pair (x,a) under policy μ, respectively, and are defined as

$$\begin{aligned}
\widetilde{d}_\gamma^\theta(x|x^0) &= (1-\gamma^2)\sum_{n=0}^{\infty} \gamma^{2n} \Pr(x_n = x | x_0 = x^0; \theta), \\
\widetilde{\pi}_\gamma^\theta(x,a|x^0) &= \widetilde{d}_\gamma^\theta(x|x^0)\mu(a|x).
\end{aligned}$$

Proof The proof of $\nabla V^\theta(x^0)$ is standard and can be found, for instance, in [44]. To prove $\nabla U^\theta(x^0)$, we start by the fact that from (2) we have $U(x) = \sum_a \mu(x|a) W(x,a)$. If we take the derivative w.r.t. θ from both sides of this equation and obtain

[3] Henceforth, we shall drop the subscript θ and use $\nabla L(\theta,\lambda)$ to denote the derivative w.r.t. θ.

$$\begin{aligned}
\nabla U(x^0) &= \sum_a \nabla\mu(a|x^0)W(x^0,a) + \sum_a \mu(a|x^0)\nabla W(x^0,a)\\
&= \sum_a \nabla\mu(a|x^0)W(x^0,a) + \sum_a \mu(a|x^0)\nabla\Big[r(x^0,a)^2 + \gamma^2 \sum_{x'} P(x'|x^0,a)U(x')\\
&+ 2\gamma r(x^0,a)\sum_{x'} P(x'|x^0,a)V(x')\Big]\\
&= \underbrace{\sum_a \nabla\mu(a|x^0)W(x^0,a) + 2\gamma \sum_{a,x'} \mu(a|x^0)r(x^0,a)P(x'|x^0,a)\nabla V(x')}_{h(x^0)}\\
&+ \gamma^2 \sum_{a,x'} \mu(a|x^0)P(x'|x^0,a)\nabla U(x')\\
&= h(x^0) + \gamma^2 \sum_{a,x'} \mu(a|x^0)P(x'|x^0,a)\nabla U(x') \qquad (5)\\
&= h(x^0) + \gamma^2 \sum_{a,x'} \mu(a|x^0)P(x'|x^0,a)\nabla\Big[h(x')\\
&\quad + \gamma^2 \sum_{a',x''} \mu(a'|x')P(x''|x',a')\nabla U(x'')\Big].
\end{aligned}$$

By unrolling the last equation using the definition of $\nabla U(x)$ from (5), we obtain

$$\begin{aligned}
\nabla U(x^0) &= \sum_{n=0}^{\infty} \gamma^{2n} \sum_x \Pr(x_n = x|x_0 = x^0)h(x) = \frac{1}{1-\gamma^2}\sum_x \widetilde{d}_\gamma(x|x^0)h(x)\\
&= \frac{1}{1-\gamma^2}\Big[\sum_{x,a} \widetilde{d}_\gamma(x|x^0)\mu(a|x)\nabla\log\mu(a|x)W(x,a)\\
&+ 2\gamma \sum_{x,a,x'} \widetilde{d}_\gamma(x|x^0)\mu(a|x)r(x,a)P(x'|x,a)\nabla V(x')\Big]\\
&= \frac{1}{1-\gamma^2}\Big[\sum_{x,a} \widetilde{\pi}_\gamma(x,a|x^0)\nabla\log\mu(a|x)W(x,a)\\
&+ 2\gamma \sum_{x,a,x'} \widetilde{\pi}_\gamma(x,a|x^0)r(x,a)P(x'|x,a)\nabla V(x')\Big].
\end{aligned}$$

■

In [66], a policy gradient result analogous to Lemma 1 is provided for the value function in the case of full-state representations. In the average reward setting, a similar result helps in extension to incorporate function approximation - see the actor-critic algorithms in [14][4]. However, a similar approach is not viable for discounted setting and this motivates the use of stochastic optimization techniques like SPSA/SF (cf. [10]). The problem is further complicated in the variance-constrained setting that we consider because:

1. two different sampling distributions, π_γ^θ and $\widetilde{\pi}_\gamma^\theta$, are used for $\nabla V^\theta(x^0)$ and $\nabla U^\theta(x^0)$, and

[4] We extend this to the case of variance-constrained MDP in Section 6.

2. $\nabla V^\theta(x')$ appears in the second sum of $\nabla U^\theta(x^0)$ equation, which implies that we need to estimate the gradient of the value function V^θ at every state of the MDP, and not just at the initial state x^0.

To alleviate the above mentioned problems, we borrow the principle of simultaneous perturbation for estimating the gradient $\nabla L(\theta, \lambda)$ and develop novel risk-sensitive actor-critic algorithms in the following section.

4 Discounted Reward Risk-Sensitive Actor-Critic Algorithms

In this section, we present actor-critic algorithms for optimizing the risk-sensitive measure (3). These algorithms are based on two simultaneous perturbation methods: *simultaneous perturbation stochastic approximation* (SPSA) and *smoothed functional* (SF).

4.1 Algorithm Structure

For the purpose of finding an optimal risk-sensitive policy, a standard procedure would update the policy parameter θ and Lagrange multiplier λ in two nested loops as follows:

- An inner loop that descends in θ using the gradient of the Lagrangian $L(\theta, \lambda)$ w.r.t. θ, and
- An outer loop that ascends in λ using the gradient of the Lagrangian $L(\theta, \lambda)$ w.r.t. λ.

Using two-timescale stochastic approximation [21, Chapter 6], the two loops above can run in parallel, as follows:

$$\theta_{n+1} = \Gamma\big[\theta_n - \zeta_2(n) A_n^{-1} \nabla L(\theta_n, \lambda_n)\big], \tag{6}$$

$$\lambda_{n+1} = \Gamma_\lambda\big[\lambda_n + \zeta_1(n) \nabla_\lambda L(\theta_n, \lambda_n)\big], \tag{7}$$

In the above,

- A_n is a positive definite matrix that fixes the order of the algorithm. For the first order methods, $A_n = I$ (I is the identity matrix), while for the second order methods $A_n \to \nabla^2_\theta L(\theta_n, \lambda_n)$ as $n \to \infty$.
- Γ is a projection operator that keeps the iterate θ_n stable by projecting onto a compact and convex set $\Theta := \prod_{i=1}^{\kappa_1}[\theta^{(i)}_{\min}, \theta^{(i)}_{\max}]$. In particular, for any $\theta \in \mathbb{R}^{\kappa}_1$, $\Gamma(\theta) = (\Gamma^{(1)}(\theta^{(1)}), \dots, \Gamma^{(\kappa_1)}(\theta^{(\kappa_1)}))^T$, with $\Gamma^{(i)}(\theta^{(i)}) := \min(\max(\theta^{(i)}_{\min}, \theta^{(i)}), \theta^{(i)}_{\max})$.
- Γ_λ is a projection operator that keeps the Lagrange multiplier λ_n within the interval $[0, \lambda_{\max}]$, for some large positive constant $\lambda_{\max} < \infty$ and can be defined in an analogous fashion as Γ.
- $\zeta_1(n), \zeta_2(n)$ are step-sizes selected such that θ update is on the faster and λ update is on the slower timescale. Note that another timescale $\zeta_3(n)$ that is the fastest is used for the TD-critic, which provides the estimate of the Lagrangian for a given (θ, λ).

Simulation optimization. We operate in a setting where we only observe simulated rewards of the underlying MDP. Thus, it is required to estimate the mean and variance of the return (we use a TD-critic for this purpose) and then use these estimates to compute gradient of the Lagrangian. The gradient $\nabla_\lambda L(\theta, \lambda)$ has a particularly simple form of $(\Lambda^\theta(x^0) - \alpha)$, suggesting the usage of sample variance constraints to perform the dual ascent for Lagrange

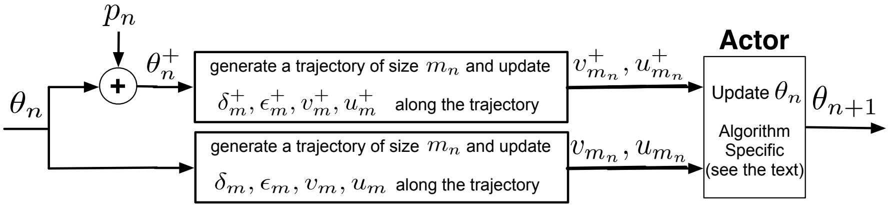

p_n
θ_n^+
θ_n
generate a trajectory of size m_n and update
$\delta_m^+, \epsilon_m^+, v_m^+, u_m^+$ along the trajectory
generate a trajectory of size m_n and update
$\delta_m, \epsilon_m, v_m, u_m$ along the trajectory
$v_{m_n}^+, u_{m_n}^+$
v_{m_n}, u_{m_n}
Actor
Update θ_n
Algorithm Specific (see the text)
θ_{n+1}

Fig. 1 The overall flow of our simultaneous perturbation based actor-critic algorithms.

multiplier λ. On the other hand, the expression for gradient of $L(\theta, \lambda)$ w.r.t. θ is complicated (see Lemma 1) and warrants the usage of a simulation optimization that can provide gradient estimates from sample observation. We employ simultaneous perturbation schemes for estimating the gradient (and in the case of second order methods, the Hessian) of the Lagrangian $L(\theta, \lambda)$. The idea in these methods is to estimate the gradients $\nabla V^\theta(x^0)$ and $\nabla U^\theta(x^0)$ (needed for estimating the gradient $\nabla L(\theta, \lambda)$) using two simulated trajectories of the system corresponding to policies with parameters θ_n and $\theta_n^+ = \theta_n + p_n$. Here p_n is a perturbation vector that is specific to the algorithm.

Based on the order, our algorithms can be classified as:

1. **First order**: This corresponds to $A_n = I$ in (6). The proposed algorithms here include RS-SPSA-G and RS-SF-G, where the former estimates the gradient using SPSA, while the latter uses SF. These algorithms use the following choice for the perturbation vector: $p_n = \beta_n \Delta_n$. Here $\beta_n > 0$ is a positive constant and Δ_n is a perturbation random variable, i.e., a κ_1-vector of independent Rademacher (for SPSA) and Gaussian $\mathcal{N}(0,1)$ (for SF) random variables.
2. **Second order**: This corresponds to A_n which converges to $\nabla^2 L(\theta_n, \lambda_n)$ as $n \to \infty$. The proposed algorithms here include RS-SPSA-N and RS-SF-N, where the former uses SPSA for gradient/Hessian estimates and the latter employs SF for the same. These algorithms use the following choice for perturbation vector: For RS-SPSA-N, $p_n = \beta_n \Delta_n + \beta_n \widehat{\Delta}_n$, $\beta_n > 0$ is a positive constant and Δ_n and $\widehat{\Delta}_n$ are perturbation parameters that are κ_1-vectors of independent Rademacher random variables, respectively. For RS-SF-N, $p_n = \beta_n \Delta_n$, where Δ_n is a κ_1 vector of Gaussian $\mathcal{N}(0,1)$ random variables.

The overall flow of our proposed actor-critic algorithms is illustrated in Figure 1 and Algorithm 1. The overall operation involves the following two loops: At each time instant n,

Inner Loop (Critic Update): For a fixed policy (given as θ_n), simulate two system trajectories, each of length m_n, as follows:

1) Unperturbed Simulation: For $m = 0, 1, \dots, m_n$, take action $a_m \sim \mu(\cdot | x_m; \theta_n)$, observe the reward $R(x_m, a_m)$, and the next state x_{m+1} in the first trajectory.

2) Perturbed Simulation: For $m = 0, 1, \dots, m_n$, take action $a_m^+ \sim \mu(\cdot | x_m^+; \theta_n^+)$, observe the reward $R(x_m^+, a_m^+)$, and the next state x_{m+1}^+ in the second trajectory.

Using the method of temporal differences (TD) [62], estimate the value functions $\widehat{V}^{\theta_n}(x^0)$ and $\widehat{V}^{\theta_n^+}(x^0)$, and square value functions $\widehat{U}^{\theta_n}(x^0)$ and $\widehat{U}^{\theta_n^+}(x^0)$, corresponding to the policy parameter θ_n and θ_n^+.

Algorithm 1 Template of the Risk-Sensitive Discounted Reward Actor-Critic Algorithms

```
Input: parameterized policy $\mu(\cdot|\cdot;\theta)$ and value function feature vectors $\phi_v(\cdot)$ and $\phi_u(\cdot)$
Initialization: policy parameter $\theta = \theta_0$; value function weight vectors $v = v_0$ and $v^+ = v_0^+$; square value function weight vectors $u = u_0$ and $u^+ = u_0^+$; initial state $x_0 \sim P_0(x)$
for $n = 0, 1, 2, \dots$ do
    for $m = 0, 1, 2, \dots, m_n$ do
        Draw action $a_m \sim \mu(\cdot|x_m;\theta_n)$, observe next state $x_{m+1}$ and reward $R(x_m, a_m)$
        Draw action $a_m^+ \sim \mu(\cdot|x_m^+;\theta_n^+)$, observe next state $x_{m+1}^+$ and reward $R(x_m^+, a_m^+)$
        Critic Update:   see (13) and (15) in the text
    end for
    Actor Update:   Algorithm-Specific
    Lagrange Multiplier Update:   see (21) in the text
end for
return policy and value function parameters $\theta, \lambda, v, u$
```

Outer Loop (Actor Update): Estimate the gradient/Hessian of $\widehat{V}^\theta(x^0)$ and $\widehat{U}^\theta(x^0)$, and hence the gradient/Hessian of Lagrangian $L(\theta, \lambda)$, using either SPSA (17) or SF (18) methods. Using these estimates, update the policy parameter θ in the descent direction using either a gradient or a Newton decrement, and the Lagrange multiplier λ in the ascent direction.

In the next section, we describe the TD-critic and subsequently, in Sections 4.3–4.4, present the first and second order actor critic algorithms, respectively.

4.2 TD-Critic

In our actor-critic algorithms, the critic uses linear approximation for the value and square value functions, i.e., $\widehat{V}(x) \approx v^\top \phi_v(x)$ and $\widehat{U}(x) \approx u^\top \phi_u(x)$, where the features $\phi_v(\cdot)$ and $\phi_u(\cdot)$ are from low-dimensional spaces $\mathbb{R}^{\kappa_2}$ and $\mathbb{R}^{\kappa_3}$, respectively. Let Φ_v and Φ_u denote $|\mathcal{X}| \times \kappa_2$ and $|\mathcal{X}| \times \kappa_3$ dimensional matrices, whose ith columns are $\phi_v^{(i)} = \big(\phi_v^{(i)}(x),\ x \in \mathcal{X}\big)^\top,\ i = 1, \dots, \kappa_2$ and $\phi_u^{(i)} = \big(\phi_u^{(i)}(x),\ x \in \mathcal{X}\big)^\top,\ i = 1, \dots, \kappa_3$. Let $S_v := \{\Phi_v v \mid v \in \mathbb{R}_2^\kappa\}$ and $S_u := \{\Phi_u u \mid u \in \mathbb{R}_3^\kappa\}$, denote the subspaces within which we approximate the value and square value functions. We make the following standard assumption as in [14]:

(A3) *The basis functions $\{\phi_v^{(i)}\}_{i=1}^{\kappa_2}$ and $\{\phi_u^{(i)}\}_{i=1}^{\kappa_3}$ are linearly independent. In particular, $\kappa_2, \kappa_3 \ll n$ and Φ_v and Φ_u are full rank. Moreover, for every $v \in \mathbb{R}^{\kappa_2}$ and $u \in \mathbb{R}^{\kappa_3}$, $\Phi_v v \neq e$ and $\Phi_u u \neq e$, where e is the n-dimensional vector with all entries equal to one.*

Let Π_u and Π_v be operators that project onto S_v and S_u, respectively and as a consequence of the above assumption, can be defined as follows:

$$\Pi_v = \Phi_v(\Phi_v^\top \boldsymbol{D}^\theta \Phi_v)^{-1}\Phi_v^\top \boldsymbol{D}^\theta \text{ and } \Pi_u = \Phi_u(\Phi_u^\top \boldsymbol{D}^\theta \Phi_u)^{-1}\Phi_u^\top \boldsymbol{D}^\theta, \tag{8}$$

where $\boldsymbol{D}^\theta$ is a diagonal $|\mathcal{X}| \times |\mathcal{X}|$ matrix with entries $d^\theta(x)$, for each $x \in \mathcal{X}$. Recall that $d^\theta(\cdot)$ denotes the stationary distribution of the Markov chain underlying policy θ.

Let $T^\theta = [T_v^\theta; T_u^\theta]$, where T_v^θ and T_u^θ denote the Bellman operators for value and square value functions of the policy governed by parameter θ, respectively. These operators are defined as: For any $y \in \mathbb{R}^{2|\mathcal{X}|}$, let y_v and y_u denote the first and last $|\mathcal{X}|$ entries, respectively. Then

$$T^\theta y = [T^\theta_v y; T^\theta_u y], \text{ where} \tag{9}$$

$$T^\theta_v y = \boldsymbol{r}^\theta + \gamma \boldsymbol{P}^\theta y_v, \tag{10}$$

$$T^\theta_u y = \boldsymbol{R}^\theta \boldsymbol{r}^\theta + 2\gamma \boldsymbol{R}^\theta \boldsymbol{P}^\theta y_v + \gamma^2 \boldsymbol{P}^\theta y_u, \tag{11}$$

where $\boldsymbol{r}^\theta$ and $\boldsymbol{P}^\theta$ are the reward vector and the transition probability matrix of policy θ, and $\boldsymbol{R}^\theta = diag(\boldsymbol{r}^\theta)$.

Let $\varPi = \begin{pmatrix} \varPi_v & 0 \\ 0 & \varPi_u \end{pmatrix}$. Also, for any $y \in \mathbb{R}^{2|\mathcal{X}|}$, define its ν-weighted norm as

$$\|y\|_\nu = \nu \|y_v\|_{\boldsymbol{D}^\theta} + (1-\nu)\|y_u\|_{\boldsymbol{D}^\theta},$$

where $\|z\|_{\boldsymbol{D}^\theta} = \sqrt{\sum_{i=1}^{|\mathcal{X}|} d^\theta(i) z_i^2}$ for any $z \in \mathbb{R}^{|\mathcal{X}|}$.

We now claim that the projected Bellman operator $\varPi T$ is a contraction mapping w.r.t ν-weighted norm, for any policy θ.

Lemma 2 *Under (A2) and (A3), there exists a $\nu \in (0,1)$ and $\bar{\gamma} < 1$ such that*

$$\|\varPi T y - \varPi T \bar{y}\|_\nu \leq \bar{\gamma} \|y - \bar{y}\|_\nu, \forall y, \bar{y} \in \mathbb{R}^{2|\mathcal{X}|}.$$

Proof See Section 7.1. ■

Let $[\varPhi_v \bar{v}; \varPhi_u \bar{u}]$ denote the unique fixed-point of the projected Bellman operator $\varPi T$, i.e.,

$$\varPhi_v \bar{v} = \varPi_v\big(T_v(\varPhi_v \bar{v})\big), \text{ and } \varPhi_u \bar{u} = \varPi_u\big(T_u(\varPhi_u \bar{u})\big), \tag{12}$$

where $\varPi_v$ and $\varPi_u$ project into the linear spaces spanned by the columns of $\varPhi_v$ and $\varPhi_u$, respectively.

We now describe the TD algorithm that updates the critic parameters corresponding to the value and square value functions (Note that we require critic estimates for both the unperturbed as well as the perturbed policy parameters). This algorithm is an extension of the algorithm proposed by [70] to the discounted setting. Recall from Algorithm 1 that, at any instant n, the TD-critic runs two m_n length trajectories corresponding to policy parameters θ_n and $\theta_n + \delta \varDelta_n$.

Critic Update: Calculate the temporal difference (TD)-errors δ_m, δ^+_m for the value and ϵ_m, ϵ^+_m for the square value functions using (15), and update the critic parameters v_m, v^+_m for the value and u_m, u^+_m for the square value functions as follows:

Unperturbed:

$$v_{m+1} = v_m + \zeta_3(m)\delta_m \phi_v(x_m), \qquad u_{m+1} = u_m + \zeta_3(m)\epsilon_m \phi_u(x_m), \tag{13}$$

Perturbed:

$$v^+_{m+1} = v^+_m + \zeta_3(m)\delta^+_m \phi_v(x^+_m), \quad u^+_{m+1} = u^+_m + \zeta_3(m)\epsilon^+_m \phi_u(x^+_m), \tag{14}$$

where the TD-errors $\delta_m, \delta_m^+, \epsilon_m, \epsilon_m^+$ in (13) are computed as

Unperturbed:

$$\delta_m = R(x_m, a_m) + \gamma v_m^\intercal \phi_v(x_{m+1}) - v_m^\intercal \phi_v(x_m), \tag{15}$$

$$\epsilon_m = R(x_m, a_m)^2 + 2\gamma R(x_m, a_m) v_m^\intercal \phi_v(x_{m+1}) + \gamma^2 u_m^\intercal \phi_u(x_{m+1}) - u_m^\intercal \phi_u(x_m),$$

Perturbed:

$$\delta_m^+ = R(x_m^+, a_m^+) + \gamma v_m^{+\top} \phi_v(x_{m+1}^+) - v_m^{+\top} \phi_v(x_m^+), \tag{16}$$

$$\begin{aligned}\epsilon_m^+ = {} & R(x_m^+, a_m^+)^2 + 2\gamma R(x_m^+, a_m^+) v_m^{+\top} \phi_v(x_{m+1}^+) + \gamma^2 u_m^{+\top} \phi_u(x_{m+1}^+) \\ & - u_m^{+\top} \phi_u(x_m^+).\end{aligned}$$

Note that the TD-error ϵ for the square value function U comes directly from its Bellman equation (2). Theorem 2 in Section 7 establishes that the critic parameters (v_n, u_n) governed by (13) converge to the solutions $(\bar{v}, \bar{u})$ of the fixed point equation (12).

Convergence rate

Let $\nu_{\min} = \min(\nu_v, \nu_u)$, where ν_v and ν_u are minimum eigenvalues of $\Phi_v^\intercal \boldsymbol{D}^\theta \Phi_v$ and $\Phi_u^\intercal \boldsymbol{D}^\theta \Phi_u$, respectively. Recall that $\boldsymbol{D}^\theta$ denotes the stationary distribution of the underlying policy θ. From (A2), (A3) and the fact that we consider finite state-spaces, we have that $\nu_{\min} > 0$.

From recent results in [35] that provide non-asymptotic bounds for TD(0) with function approximation, we know that the canonical $O(m^{-1/2})$ rate can be achieved under the appropriate choice of the step-size $\zeta_3(m)$. The following rate result is crucial in setting the trajectory lengths m_n and relating them to perturbation constants β_n (see (A4) in the next section):

Theorem 1 *Under (A2)-(A3), choosing* $\zeta_3(m) = \frac{c_0 c}{(c+m)}$*, with* $c_0 < \nu_{\min}(1-\gamma)/(2(1+\gamma)^2)$ *and c such that* $\nu_{\min}(1-\gamma)c_0 c > 1$*, we have,*

$$\mathbb{E}\left\|v_m - \bar{v}\right\|_2 \leq \frac{K_1(m)}{\sqrt{m+c}} \quad \textit{and} \quad \mathbb{E}\left\|u_m - \bar{u}\right\|_2 \leq \frac{K_2(m)}{\sqrt{m+c}},$$

where $K_1(m)$ *and* $K_2(m)$ *are* $O(1)$.

Proof The first claim follows directly from Theorem 1 in [35], while the second claim can be proven in an analogous manner as the first.

The above rate result holds only if the step-size is set using $\nu_{\min}$ and the latter quantity is unknown in a typical RL setting. However, a standard trick to overcome this dependence while obtaining the same convergence rate is to employ iterate averaging, proposed independently by Polyak [45] and Ruppert [51]. The latter approach involves using a larger step-size $\Theta(1/n^{\varsigma_1})$ with $\varsigma_1 \in (1/2, 1)$ and couple this with averaging of iterates. An iterate averaged variant of Theorem 1 can be claimed and we refer the reader to Theorem 2 of [35] for further details.

4.3 First-Order Algorithms: RS-SPSA-G and RS-SF-G

SPSA-based estimate for $\nabla V^\theta(x^0)$, and similarly for $\nabla U^\theta(x^0)$, is given by

$$\nabla_i \widehat{V}^{\theta_n}(x^0) \quad \approx \quad \frac{\widehat{V}^{\theta_n+\beta_n\Delta_n}(x^0) - \widehat{V}^{\theta_n}(x^0)}{\beta_n \Delta^{(i)}}, \qquad i = 1,\ldots,\kappa_1, \tag{17}$$

where β_n are perturbation constants that vanish asymptotically (see (A4) at the end of this section) and Δ_n is a vector of independent Rademacher random variables, for all $n = 1,2,\ldots$. The advantage of this estimator is that it perturbs all directions at the same time (the numerator is identical in all κ_1 components). So, the number of function measurements needed for this estimator is always two, independent of the dimension κ_1. However, unlike the SPSA estimates in [58] that use two-sided balanced estimates (simulations with parameters $\theta_n - \beta_n\Delta_n$ and $\theta + \beta\Delta$), our gradient estimates are one-sided (simulations with parameters θ_n and $\theta_n + \beta_n\Delta_n$) and resemble those in [24]. The use of one-sided estimates is primarily because the updates of the Lagrangian parameter require a simulation with the running parameter θ_n. Using a balanced gradient estimate would therefore come at the cost of an additional simulation (the resulting procedure would then require three simulations), which we avoid by using one-sided gradient estimates.

SF-based method estimates not the gradient of a function $H(\theta_n)$ itself, but rather the convolution of $\nabla H(\theta_n)$ with the Gaussian density function $\mathcal{N}(\mathbf{0}, \beta_n^2 \boldsymbol{I})$, i.e.,

$$\begin{aligned} C_{\beta_n} H(\theta_n) &= \int \mathcal{G}_{\beta_n}(\theta_n - z)\nabla_z H(z)dz = \int \nabla_z \mathcal{G}_{\beta_n}(z) H(\theta_n - z)dz \\ &= \frac{1}{\beta_n}\int -z'\mathcal{G}_1(z')H(\theta_n - \beta_n z')dz', \end{aligned}$$

where $\mathcal{G}_{\beta_n}$ is the κ_1-dimensional Gaussian p.d.f. The first equality above follows by using integration by parts and the second one by using the fact that $\nabla_z \mathcal{G}_{\beta_n}(z) = \frac{-z}{\beta_n^2}\mathcal{G}_{\beta_n}(z)$ and by substituting $z' = z/\beta_n$. As $\beta_n \to 0$, it can be seen that $C_{\beta_n} H(\theta_n)$ converges to $\nabla H(\theta_n)$ (see Chapter 6 of [17]). Thus, a one-sided SF estimate of $\nabla V^{\theta_n}(x^0)$ is given by

$$\nabla_i \widehat{V}^{\theta_n}(x^0) \quad \approx \quad \frac{\Delta_n^{(i)}}{\beta_n}\left(\widehat{V}^{\theta_n+\beta_n\Delta_n}(x^0) - \widehat{V}^{\theta_n}(x^0)\right), \qquad i = 1,\ldots,\kappa_1, \tag{18}$$

where Δ_n is a vector of independent Gaussian $\mathcal{N}(0,1)$ random variables. The reasons for using the one-sided estimate in (18) are as follows: (i) the estimate in (18) has lower bias when compared to a one simulation estimate that does not use $\widehat{V}^{\theta_n}(x^0)$ and (ii) for updating the Lagrange multiplier λ, we require a trajectory of the MDP corresponding to policy θ_n and this trajectory can be used to estimate $\widehat{V}^{\theta_n}(x^0)$.

Actor Update: Estimate the gradients $\nabla V^\theta(x^0)$ and $\nabla U^\theta(x^0)$ using SPSA (17) or SF (18) and update the policy parameter θ as follows[5]: For $i = 1, \ldots, \kappa_1$,

RS-SPSA-G:

$$\theta_{n+1}^{(i)} = \Gamma_i \Bigg[\theta_n^{(i)} + \frac{\zeta_2(n)}{\beta_n \Delta_n^{(i)}} \Big(\big(1 + 2\lambda_n v_n^\intercal \phi_v(x^0)\big)(v_n^+ - v_n)^\intercal \phi_v(x^0) \\ - \lambda_n (u_n^+ - u_n)^\intercal \phi_u(x^0) \Big) \Bigg], \tag{19}$$

RS-SF-G:

$$\theta_{n+1}^{(i)} = \Gamma_i \Bigg[\theta_n^{(i)} + \frac{\zeta_2(n) \Delta_n^{(i)}}{\beta_n} \Big(\big(1 + 2\lambda_n v_n^\intercal \phi_v(x^0)\big)(v_n^+ - v_n)^\intercal \phi_v(x^0) \\ - \lambda_n (u_n^+ - u_n)^\intercal \phi_u(x^0) \Big) \Bigg]. \tag{20}$$

For both SPSA and SF variants, the Lagrange multiplier λ is updated as follows:

$$\lambda_{n+1} = \Gamma_\lambda \bigg[\lambda_n + \zeta_1(n) \Big(u_n^\intercal \phi_u(x^0) - \big(v_n^\intercal \phi_v(x^0)\big)^2 - \alpha \Big) \bigg]. \tag{21}$$

In the above, note the following:

(i) $\beta_n \geq 0$ and vanish asymptotically (see (A4) below for the precise condition);

(ii) $\Delta_n^{(i)}$'s are independent Rademacher and Gaussian $\mathcal{N}(0,1)$ random variables in SPSA and SF updates, respectively;

(iii) Γ and Γ_λ are projection operators that keep the iterates (θ_n, λ_n) stable and were defined in Section 4.1. These projection operators are necessary to keep the iterates stable and hence, ensure convergence of the algorithms.

Choosing trajectory length m_n, perturbation constants β_n and step-sizes $\zeta_3(n), \zeta_2(n), \zeta_1(n)$

We make the following assumption on the step-size schedules:

(A4) The step size schedules $\{\zeta_2(n)\}$, and $\{\zeta_1(n)\}$ satisfy

$$\zeta_2(n), \beta_n \to 0, \frac{1}{\sqrt{m_n}\beta_n} \to 0, \tag{22}$$

$$\sum_n \zeta_1(n) = \sum_n \zeta_2(n) = \infty, \tag{23}$$

$$\sum_n \zeta_1(n)^2, \quad \sum_n \frac{\zeta_2(n)^2}{\beta_n^2}, \quad < \infty, \tag{24}$$

$$\zeta_1(n) = o\big(\zeta_2(n)\big). \tag{25}$$

Equations 23 and 24 are standard step-size conditions in stochastic approximation algorithms, and Equation 25 ensures that the policy parameter update is on the faster time-scale $\{\zeta_2(n)\}$, and the Lagrange multiplier update is on the slower time-scale $\{\zeta_1(n)\}$.

[5] By an abuse of notation, we use v_n (resp. v_n^+, u_n, u_n^+) to denote the critic parameter v_{m_n} (resp. $v_{m_n}^+, u_{m_n}, u_{m_n}^+$) obtained at the end of a m_n length trajectory.

Equation 22 is motivated by a similar condition in [49] and ensures that the bias from a finite length (m_n) trajectory run of TD-critic can be ignored. A simple setting that ensures (22) is to have $m_n = C_1 n^{\varsigma_2}$ and $\beta_n = C_2 n^{-\varsigma_3}$, where C_1, C_2 are constants and $\varsigma_2, \varsigma_3 > 0$ with $\varsigma_3 > \varsigma_2/2$. This ensures that the trajectories increase in length as a function of outer loop index n, at a rate that is sufficient to cancel the bias induced by the TD-critic. See Lemma 6 in Section 7 makes this claim precise, in particular justifying the need for (22) in (A4).

We provide a proof of convergence of the first-order SPSA and SF algorithms to a tuple $(\theta^{\lambda^*}, \lambda^*)$, which is a (local) saddle point of the risk-sensitive objective function $\widehat{L}(\theta, \lambda) \stackrel{\triangle}{=} -\widehat{V}^\theta(x^0) + \lambda(\widehat{\varLambda}^\theta(x^0) - \alpha)$, where $\widehat{V}^\theta(x^0) = \bar{v}^\top \phi_v(x^0)$ and $\widehat{\varLambda}^\theta(x^0) = \bar{u}^\top \phi_u(x^0) - (\bar{v}^\top \phi_v(x^0))^2$ with $\bar{v}$ and $\bar{u}$ defined by (12). Further, the limit θ^{λ^*} satisfies the variance constraint, i.e., $\widehat{\varLambda}^{\theta^{\lambda^*}}(x^0) \leq \alpha$. See Theorems 3–5 and Proposition 1 in Section 7 for details.

Remark 3 **(Extension to Sharpe Ratio Optimization)**

The gradient of Sharpe ratio (SR), $S(\theta)$, in the discounted setting is given by

$$\nabla S(\theta) = \frac{1}{\sqrt{\varLambda^\theta(x^0)}} \left(\nabla V^\theta(x^0) - \frac{V^\theta(x^0)}{2\varLambda^\theta(x^0)} \nabla \varLambda^\theta(x^0) \right).$$

The actor recursions for the variants of the RS-SPSA-G and RS-SF-G algorithms that optimize the SR objective are as follows:

RS-SPSA-G

$$\theta_{n+1}^{(i)} = \varGamma_i \Bigg(\theta_n^{(i)} + \frac{\zeta_2(n)}{\sqrt{u_n^\top \phi_u(x^0) - \big(v_n^\top \phi_v(x^0)\big)^2} \beta_n \varDelta_n^{(i)}} \Bigg((v_n^+ - v_n)^\top \phi_v(x^0) \tag{26}$$

$$- \frac{v_n^\top \phi_v(x^0)\big((u_n^+ - u_n)^\top \phi_u(x^0) - 2 v_n^\top \phi_v(x^0)(v_n^+ - v_n)^\top \phi_v(x^0)\big)}{2\Big(u_n^\top \phi_u(x^0) - \big(v_n^\top \phi_v(x^0)\big)^2\Big)} \Bigg)\Bigg).$$

RS-SF-G

$$\theta_{n+1}^{(i)} = \varGamma_i \Bigg(\theta_n^{(i)} + \frac{\zeta_2(n) \varDelta_n^{(i)}}{\beta_n \sqrt{u_n^\top \phi_u(x^0) - \big(v_n^\top \phi_v(x^0)\big)^2}} \Bigg((v_n^+ - v_n)^\top \phi_v(x^0) \tag{27}$$

$$- \frac{v_n^\top \phi_v(x^0)\big((u_n^+ - u_n)^\top \phi_u(x^0) - 2 v_n^\top \phi_v(x^0)(v_n^+ - v_n)^\top \phi_v(x^0)\big)}{2\Big(u_n^\top \phi_u(x^0) - \big(v_n^\top \phi_v(x^0)\big)^2\Big)} \Bigg)\Bigg).$$

Note that only the actor recursion changes for SR optimization, while the rest of the updates that include the critic recursions for nominal and perturbed parameters remain the same as before in the SPSA and SF based algorithms. Further, SR optimization does not involve the Lagrange parameter λ, and thus, the proposed actor-critic algorithms are two time-scale (instead of three time-scale as in the described algorithms) stochastic approximation algorithms in this case.

Remark 4 **(One-simulation SR variant.)** For the SR objective, the proposed algorithms can be modified to work with only one simulated trajectory of the system. This is because in the SR case, we do not require the Lagrange multiplier λ, and thus, the simulated trajectory corresponding to the nominal policy parameter θ is not necessary. In this implementation, the gradient is estimated as $\nabla_i S(\theta) \approx S(\theta + \beta\varDelta)/\beta\varDelta^{(i)}$ for SPSA and as $\nabla_i S(\theta) \approx (\varDelta^{(i)}/\beta) S(\theta + \beta\varDelta)$ for SF.

Remark 5 **(Monte-Carlo Critic)** In the above algorithms, the critic uses a TD method to evaluate the policies. These algorithms can be implemented with a Monte-Carlo critic that at each time instant n computes a sample average of the total discounted rewards corresponding to the nominal θ_n and perturbed $\theta_n + \beta\Delta_n$ policy parameter. This implementation would be similar to that in [68], except here we use simultaneous perturbation methods to estimate the gradient.

4.4 Second-Order Algorithms: RS-SPSA-N and RS-SF-N

Recall from Section 4.1 that a second-order scheme updates the policy parameter in the following manner:

$$\theta_{n+1} = \Gamma\big[\theta_n - \zeta_2(n)\nabla^2_\theta L(\theta,\lambda)^{-1}\nabla L(\theta,\lambda)\big]. \tag{28}$$

From the above, it is evident that for any second-order method, an estimate of the Hessian $\nabla^2_\theta L(\theta,\lambda)$ of the Lagrangian is necessary, in addition to an estimate of the gradient $\nabla L(\theta,\lambda)$. As in the case of the gradient based schemes outlined earlier, we employ the simultaneous perturbation technique to develop these estimates. The first algorithm, henceforth referred to as RS-SPSA-N, uses SPSA for the gradient/Hessian estimates. On the other hand, the second algorithm, henceforth referred to as RS-SF-N, uses a smoothed functional (SF) approach for the gradient/Hessian estimates. As confirmed by our numerical experiments, second order methods are in general more accurate, though at the cost of inverting the Hessian matrix in each step.

4.4.1 RS-SPSA-N Algorithm

The Hessian w.r.t. θ of $L(\theta,\lambda)$ can be written as follows:

$$\begin{aligned}\nabla^2_\theta L(\theta,\lambda) &= -\nabla^2_\theta V^\theta(x^0) + \lambda\nabla^2_\theta \Lambda^\theta(x^0) \\ &= -\nabla^2 V^\theta(x^0) + \lambda\left(\nabla^2 U^\theta(x^0) - 2V^\theta(x^0)\nabla^2 V^\theta(x^0) - 2\nabla V^\theta(x^0)\nabla V^\theta(x^0)^\mathsf{T}\right).\end{aligned} \tag{29}$$

Critic Update: As in the case of the gradient based schemes, we run two simulations. However, perturbed simulation here corresponds to the policy parameter $\theta_n + \beta_n(\Delta_n + \widehat{\Delta}_n)$, where Δ_n and $\widehat{\Delta}_n$ represent vectors of independent κ_1-dimensional Rademacher random variables. The critic parameters v_n, u_n from unperturbed simulation and v_n^+, u_n^+ from perturbed simulation are updated as described earlier in Section 4.2.

Gradient and Hessian Estimates: Using an SPSA-based estimation technique (see Chapter 7 of [17]), the gradient and Hessian of the value function V, and similarly of the square value function U, are estimated as follows: For $i = 1,\dots,\kappa_1$,

$$\begin{aligned}\nabla_i \widehat{V}^\theta(x^0) &\approx \frac{\widehat{V}^{\theta+\beta_n(\Delta+\widehat{\Delta})}(x^0) - \widehat{V}^\theta(x^0)}{\beta_n\Delta^{(i)}} = \frac{(v_n^+ - v_n)^\mathsf{T}\phi_v(x^0)}{\beta_n\Delta^{(i)}}, \\ \nabla^2_{i,j}\widehat{V}^\theta(x^0) &\approx \frac{\widehat{V}^{\theta+\beta_n(\Delta+\widehat{\Delta})}(x^0) - \widehat{V}^\theta(x^0)}{\beta_n^2\Delta^{(i)}\widehat{\Delta}^{(j)}} = \frac{(v_n^+ - v_n)^\mathsf{T}\phi_v(x^0)}{\beta_n^2\Delta^{(i)}\widehat{\Delta}^{(j)}}.\end{aligned}$$

As in the case of the first order algorithms, the TD-critic trajectory lengths are chosen such that there is no bias in the value estimates, when viewed from the actor-recursion. Next, using suitable Taylor expansions and observe that the bias terms vanish as $\Delta_n, \widehat{\Delta}_n$, being Rademacher, are zero-mean - see Lemma 7 in Section 7 for details. As in the case of RS-SPSA, this is an one-sided estimate with the unperturbed simulation required for updating the Lagrange multiplier.

Hessian Update: Using the critic values from the two simulations, we estimate the Hessian $\nabla^2_\theta L(\theta,\lambda)$ as follows: Let $H_n^{(i,j)}$ denote the nth estimate of the (i,j)th element of the Hessian. Then, for $i,j = 1,\ldots,\kappa_1$, with $i \le j$, the update is

$$H_{n+1}^{(i,j)} = H_n^{(i,j)} + \zeta_2'(n)\Bigg[\frac{\big(1+\lambda_n(v_n+v_n^+)^\intercal\phi_v(x^0)\big)(v_n-v_n^+)^\intercal\phi_v(x^0)}{\beta_n^2\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}} + \frac{\lambda_n(u_n^+-u_n)^\intercal\phi_u(x^0)}{\beta_n^2\Delta_n^{(i)}\widehat{\Delta}_n^{(j)}} - H_n^{(i,j)}\Bigg], \tag{30}$$

and for $i > j$, we simply set $H_{n+1}^{(i,j)} = H_{n+1}^{(j,i)}$. In the above, the step-size $\zeta_2'(n)$ satisfies

$$\sum_n \zeta_2'(n) = \infty; \sum_n {\zeta_2'}^2(n) < \infty, \frac{\zeta_2(n)}{\zeta_2'(n)} \to 0 \text{ as } n \to \infty.$$

The last condition above ensures that the Hessian update proceeds on a faster timescale in comparison to the θ-recursion (see (31) below). Finally, we set $H_{n+1} = \Upsilon\big([H_{n+1}^{(i,j)}]_{i,j=1}^{|\kappa_1|}\big)$, where $\Upsilon(\cdot)$ denotes an operator that projects a square matrix onto the set of symmetric and positive definite matrices. This projection is a standard requirement to ensure convergence of H_n to the Hessian $\nabla^2_\theta L(\theta,\lambda)$ and we state the following standard assumption (cf. [17, Chapter 7]) on this operator:

(A5) *For any sequence of matrices $\{A_n\}$ and $\{B_n\}$ in $\mathcal{R}^{\kappa_1\times\kappa_1}$ such that $\lim\limits_{n\to\infty} \parallel A_n - B_n \parallel = 0$, the Υ operator satisfies $\lim\limits_{n\to\infty} \parallel \Upsilon(A_n) - \Upsilon(B_n) \parallel = 0$. Further, for any sequence of matrices $\{C_n\}$ in $\mathcal{R}^{\kappa_1\times\kappa_1}$, we have*

$$\sup_n \parallel C_n \parallel < \infty \quad \Rightarrow \quad \sup_n \parallel \Upsilon(C_n) \parallel < \infty \textit{ and } \sup_n \parallel \{\Upsilon(C_n)\}^{-1} \parallel < \infty.$$

As suggested in [30], a possible definition of Υ is to perform an eigen-decomposition of H_n and then make all eigenvalues positive. This avoids singularity of H_n and also satisfies the above assumption. In our experiments, we use this scheme for projecting H_n.

Actor Update: Let $M_n \overset{\triangle}{=} H_n^{-1}$ denote the inverse of the the Hessian estimate H_n. We incorporate a Newton decrement to update the policy parameter θ as follows:

$$\theta_{n+1}^{(i)} = \Gamma_i\Bigg[\theta_n^{(i)} + \zeta_2(n)\sum_{j=1}^{\kappa_1} M_n^{(i,j)}\Big(\frac{\big(1+2\lambda_n v_n^\intercal\phi_v(x^0)\big)(v_n^+-v_n)^\intercal\phi_v(x^0)}{\beta_n\Delta_n^{(j)}} - \frac{\lambda_n(u_n^+-u_n)^\intercal\phi_u(x^0)}{\beta_n\Delta_n^{(j)}}\Big)\Bigg]. \tag{31}$$

In the long run, M_n converges to $\nabla^2_\theta L(\theta,\lambda)^{-1}$, while the last term in the brackets in (31) converges to $\nabla L(\theta,\lambda)$ and hence, the update (31) can be seen to descend in θ using a Newton decrement. Note that the Lagrange multiplier update here is the same as that in RS-SPSA-G.

4.4.2 RS-SF-N Algorithm

Gradient and Hessian Estimates: While the gradient estimate here is the same as that in the RS-SF-G algorithm, the Hessian is estimated as follows: Recall that $\Delta_n = \big(\Delta_n^{(1)}, \dots, \Delta_n^{(\kappa_1)}\big)^\mathsf{T}$ is a vector of mutually independent $\mathcal{N}(0,1)$ random variables. Let $\bar{H}(\Delta_n)$ be a $\kappa_1 \times \kappa_1$ matrix defined as

$$\bar{H}(\Delta_n) \triangleq \begin{bmatrix} \big(\Delta_n^{(1)^2} - 1\big) & \Delta_n^{(1)}\Delta_n^{(2)} & \cdots & \Delta_n^{(1)}\Delta_n^{(\kappa_1)} \\ \Delta_n^{(2)}\Delta_n^{(1)} & \big(\Delta_n^{(2)^2} - 1\big) & \cdots & \Delta_n^{(2)}\Delta_n^{(\kappa_1)} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta_n^{(\kappa_1)}\Delta_n^{(1)} & \Delta_n^{(\kappa_1)}\Delta_n^{(2)} & \cdots & \big(\Delta_n^{(\kappa_1)^2} - 1\big) \end{bmatrix}. \tag{32}$$

Then, the Hessian $\nabla_\theta^2 L(\theta, \lambda)$ is approximated as

$$\nabla_\theta^2 L(\theta, \lambda) \approx \frac{1}{\beta_n^2}\Big[\bar{H}(\Delta)\big(L(\theta + \beta\Delta, \lambda) - L(\theta, \lambda)\big)\Big]. \tag{33}$$

The correctness of the above estimate in the limit as $\beta_n \to 0$ can be seen from Lemma 8 in the Appendix. The main idea involves convolving the Hessian with a Gaussian density function (similar to RS-SF) and then performing integration by parts twice.

Critic Update: As in the case of the RS-SF-G algorithm, we run two simulations with unperturbed and perturbed policy parameters, respectively. Recall that the perturbed simulation corresponds to the policy parameter $\theta_n + \beta_n\Delta_n$, where Δ_n represent a vector of independent κ_1-dimensional Gaussian $\mathcal{N}(0,1)$ random variables. The critic parameters for both these simulations are updated as described earlier in Section 4.2.

Hessian Update: As in RS-SPSA-N, let $H_n^{(i,j)}$ denote the (i,j)th element of the Hessian estimate H_n at time step t. Using (33), we devise the following update rule for the Hessian estimate H_n: For $i, j, k = 1, \dots, \kappa_1$, $j < k$, the update is

$$\begin{aligned} H_{t+1}^{(i,i)} = H_n^{(i,i)} + \zeta_2'(n)\Bigg[\frac{\big(\Delta_n^{(i)^2} - 1\big)}{\beta_n^2}\Big(\big(1 + \lambda_n(v_n + v_n^+)^\mathsf{T}\phi_v(x^0)\big)(v_n - v_n^+)^\mathsf{T}\phi_v(x^0) \\ + \lambda_n(u_n^+ - u_n)^\mathsf{T}\phi_u(x^0)\Big) - H_n^{(i,i)}\Bigg], \end{aligned} \tag{34}$$

$$\begin{aligned} H_{t+1}^{(j,k)} = H_n^{(j,k)} + \zeta_2'(n)\Bigg[\frac{\Delta_n^{(j)}\Delta_n^{(k)}}{\beta_n^2}\Big(\big(1 + \lambda_n(v_n + v_n^+)^\mathsf{T}\phi_v(x^0)\big)(v_n - v_n^+)^\mathsf{T}\phi_v(x^0) \\ + \lambda_n(u_n^+ - u_n)^\mathsf{T}\phi_u(x^0)\Big) - H_n^{(j,k)}\Bigg], \end{aligned} \tag{35}$$

and for $j > k$, we set $H_{n+1}^{(j,k)} = H_{n+1}^{(k,j)}$. The step-size $\zeta_2'(n)$ is as in RS-SPSA-N. Further, as in the latter algorithm, we set $H_{n+1} = \Upsilon\big([H_{n+1}^{(i,j)}]_{i,j=1}^{|\kappa_1|}\big)$ and let $M_{n+1} \triangleq H_{n+1}^{-1}$ denote its inverse.

Actor Update: Using the gradient and Hessian estimates from the above, we update the policy parameter θ as follows:

$$\theta_{n+1}^{(i)} = \Gamma_i\Bigg[\theta_n^{(i)} + \zeta_2(n)\sum_{j=1}^{\kappa_1} M_n^{(i,j)}\frac{\Delta_n^{(j)}}{\beta_n}\Big(\big(1+2\lambda_n v_n^\intercal\phi_v(x^0)\big)(v_n^+ - v_n)^\intercal\phi_v(x^0) \\ - \lambda_n(u_n^+ - u_n)^\intercal\phi_u(x^0)\Big)\Bigg]. \quad (36)$$

As in the case of RS-SPSA-N, it can be seen that the above update rule is equivalent to descent with a Newton decrement, since M_n converges to $\nabla_\theta^2 L(\theta,\lambda)^{-1}$, and the last term in the brackets in (36) converges to $\nabla L(\theta,\lambda)$. The Lagrange multiplier λ update here is the same as that in RS-SF-G.

Remark 6 The second-order variants of the algorithms for SR optimization can be worked out along similar lines as outlined in Section 4.4 and the details are omitted here.

5 Average Reward Setting

The average reward under policy μ is defined as

$$\rho(\mu) \;=\; \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{n=0}^{T-1} R_n \mid \mu\right] \;=\; \sum_{x,a} d^\mu(x)\mu(a|x)r(x,a) \;=\; \sum_{x,a}\pi^\mu(x,a)r(x,a),$$

where d^μ and π^μ are the stationary distributions of policy μ over states and state-action pairs, respectively (see Section 2). The goal in the standard (risk-neutral) average reward formulation is to find an *average optimal* policy, i.e., $\mu^* = \arg\max_\mu \rho(\mu)$. For all states $x \in \mathcal{X}$ and actions $a \in \mathcal{A}$, the *differential* action-value and value functions of policy μ are defined respectively as

$$Q^\mu(x,a) = \sum_{n=0}^{\infty}\mathbb{E}\big[R_n - \rho(\mu) \mid x_0 = x, a_0 = a, \mu\big],$$
$$V^\mu(x) = \sum_a \mu(a|x)Q^\mu(x,a).$$

These functions satisfy the following Poisson equations [50]

$$\rho(\mu) + V^\mu(x) = \sum_a \mu(a|x)\big[r(x,a) + \sum_{x'} P(x'|x,a)V^\mu(x')\big], \quad (37)$$

$$\rho(\mu) + Q^\mu(x,a) = r(x,a) + \sum_{x'} P(x'|x,a)V^\mu(x'). \quad (38)$$

In the context of risk-sensitive MDPs, different criteria have been proposed to define a measure of *variability* in the average reward setting, among which we consider the *long-run variance* of μ [28] defined as

$$\Lambda(\mu) \;=\; \sum_{x,a}\pi^\mu(x,a)\big[r(x,a)-\rho(\mu)\big]^2 \;=\; \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{n=0}^{T-1}\big(R_n-\rho(\mu)\big)^2\Big|\,\mu\right]. \quad (39)$$

This notion of variability is based on the observation that it is the frequency of occurrence of state-action pairs that determine the variability in the average reward. It is easy to show that

$$\Lambda(\mu) = \eta(\mu) - \rho(\mu)^2, \qquad \text{where} \qquad \eta(\mu) = \sum_{x,a} \pi^\mu(x,a) r(x,a)^2.$$

We consider the following risk-sensitive measure for average reward MDPs in this paper:

$$\max_\theta \rho(\theta) \qquad \text{subject to} \qquad \Lambda(\theta) \leq \alpha, \tag{40}$$

for a given $\alpha > 0$.[6] As in the discounted setting, we employ the Lagrangian relaxation procedure to convert (40) to the unconstrained problem

$$\max_\lambda \min_\theta \Big(L(\theta,\lambda) \overset{\triangle}{=} -\rho(\theta) + \lambda\big(\Lambda(\theta) - \alpha\big)\Big).$$

As in the discounted setting, we descend in θ using $\nabla L(\theta,\lambda) = -\nabla\rho(\theta) + \lambda\nabla\Lambda(\theta)$ and ascend in λ using $\nabla_\lambda L(\theta,\lambda) = \Lambda(\theta) - \alpha$, to find the saddle point of $L(\theta,\lambda)$. Since $\nabla\Lambda(\theta) = \nabla\eta(\theta) - 2\rho(\theta)\nabla\rho(\theta)$, in order to compute $\nabla\Lambda(\theta)$ it would be enough to calculate $\nabla\rho(\theta)$ and $\nabla\eta(\theta)$. Let U^μ and W^μ denote the differential value and action-value functions associated with the square reward under policy μ, respectively. These two quantities satisfy the following Poisson equations:

$$\eta(\mu) + U^\mu(x) = \sum_a \mu(a|x)\big[r(x,a)^2 + \sum_{x'} P(x'|x,a)U^\mu(x')\big],$$
$$\eta(\mu) + W^\mu(x,a) = r(x,a)^2 + \sum_{x'} P(x'|x,a)U^\mu(x'). \tag{41}$$

The gradients of $\rho(\theta)$ and $\eta(\theta)$ are given by the following lemma:

Lemma 3 *Under (A1) and (A2), we have*

$$\nabla\rho(\theta) = \sum_{x,a} \pi^\theta(x,a)\nabla\log\mu(a|x;\theta)Q(x,a;\theta), \tag{42}$$
$$\nabla\eta(\theta) = \sum_{x,a} \pi^\theta(x,a)\nabla\log\mu(a|x;\theta)W(x,a;\theta). \tag{43}$$

Proof The proof of $\nabla\rho(\theta)$ can be found in the literature (e.g., [65, 33]). To prove $\nabla\eta(\theta)$, we start by the fact that from (41), we have $U(x) = \sum_a \mu(x|a)W(x,a)$. If we take the derivative w.r.t. θ from both sides of this equation, we obtain

$$\begin{aligned}\nabla U(x) &= \sum_a \nabla\mu(x|a)W(x,a) + \sum_a \mu(x|a)\nabla W(x,a)\\ &= \sum_a \nabla\mu(x|a)W(x,a) + \sum_a \mu(x|a)\nabla\big(r(x,a)^2 - \eta + \sum_{x'} P(x'|x,a)U(x')\big)\\ &= \sum_a \nabla\mu(x|a)W(x,a) - \nabla\eta + \sum_{a,x'} \mu(a|x)P(x'|x,a)\nabla U(x').\end{aligned} \tag{44}$$

[6] Similar to the discounted setting, the risk-sensitive average reward algorithm proposed in this paper can be easily extended to other risk measures based on the long-term variance of μ, including the Sharpe Ratio (SR), i.e., $\max_\theta \rho(\theta)/\sqrt{\Lambda(\theta)}$. The extension to SR will be described in more details in Section 7.

The second equality is by replacing $W(x,a)$ from (41). Now if we take the weighted sum, weighted by $d^\mu(x) = \boldsymbol{D}^\theta(x)$, from both sides of (44), we have

$$\sum_x d^\mu(x)\nabla U(x) = \sum_{x,a} d^\mu(x)\nabla\mu(a|x)W(x,a) - \nabla\eta \\ + \sum_{x,a,x'} d^\mu(x)\mu(a|x)P(x'|x,a)\nabla U(x'). \tag{45}$$

The claim follows from the fact that the last sum on the RHS of (45) is equal to $\sum_x d^\mu(x)\nabla U(x)$. ■

Note that (43) for calculating $\nabla\eta(\theta)$ has close resemblance to (42) for $\nabla\rho(\theta)$, and thus, similar to what we have for (42), any function $b : \mathcal{X} \to \mathbb{R}$ can be added or subtracted to $W(x,a;\theta)$ on the RHS of (43) without changing the result of the integral (see e.g., [14]). So, we can replace $W(x,a;\theta)$ with the square reward advantage function $B(x,a;\theta) = W(x,a;\theta) - U(x;\theta)$ on the RHS of (43) in the same manner as we can replace $Q(x,a;\theta)$ with the advantage function $A(x,a;\theta) = Q(x,a;\theta) - V(x;\theta)$ on the RHS of (42) without changing the result of the integral. We define the temporal difference (TD) errors δ_n and ϵ_n for the differential value and square value functions as

$$\delta_n = R(x_n,a_n) - \widehat{\rho}_{n+1} + \widehat{V}(x_{n+1}) - \widehat{V}(x_n),$$
$$\epsilon_n = R(x_n,a_n)^2 - \widehat{\eta}_{n+1} + \widehat{U}(x_{n+1}) - \widehat{U}(x_n).$$

If $\widehat{V}$, $\widehat{U}$, $\widehat{\rho}$, and $\widehat{\eta}$ are unbiased estimators of V^μ, U^μ, $\rho(\mu)$, and $\eta(\mu)$, respectively, then we show in Lemma 4 that δ_n and ϵ_n are unbiased estimates of the advantage functions A^μ and B^μ, i.e., $\mathbb{E}[\,\delta_n|\,x_n,a_n,\mu] = A^\mu(x_n,a_n)$ and $\mathbb{E}[\,\epsilon_n|\,x_n,a_n,\mu] = B^\mu(x_n,a_n)$.

Lemma 4 *For any given policy μ, we have*

$$\mathbb{E}[\,\delta_n|\,x_n,a_n,\mu] = A^\mu(x_n,a_n), \qquad \mathbb{E}[\,\epsilon_n|\,x_n,a_n,\mu] = B^\mu(x_n,a_n).$$

Proof The first statement $\mathbb{E}[\,\delta_n|\,x_n,a_n,\mu] = A^\mu(x_n,a_n)$ has been proved in Lemma 3 of [14], so here we only prove the second statement $\mathbb{E}[\,\epsilon_n|\,x_n,a_n,\mu] = B^\mu(x_n,a_n)$. we may write

$$\begin{aligned}
\mathbb{E}[\,\epsilon_n|\,x_n,a_n,\mu] &= \mathbb{E}\big[R(x_n,a_n)^2 - \widehat{\eta}_{n+1} + \widehat{U}(x_{n+1}) - \widehat{U}(x_n) \mid x_n,a_n,\mu\big] \\
&= r(x_n,a_n)^2 - \eta(\mu) + \mathbb{E}\big[\widehat{U}(x_{n+1}) \mid x_n,a_n,\mu\big] - U^\mu(x_n) \\
&= r(x_n,a_n)^2 - \eta(\mu) + \mathbb{E}\Big[\mathbb{E}\big[\widehat{U}(x_{n+1}) \mid x_{n+1},\mu\big] \mid x_n,a_n\Big] - U^\mu(x_n) \\
&= r(x_n,a_n)^2 - \eta(\mu) + \mathbb{E}\big[\widehat{U}(x_{n+1}) \mid x_n,a_n\big] - U^\mu(x_n) \\
&= \underbrace{r(x_n,a_n)^2 - \eta(\mu) + \sum_{x_{n+1}\in\mathcal{X}} P(x_{n+1}|x_n,a_n)U^\mu(x_{n+1})}_{W^\mu(x,a)} - U^\mu(x_n) \\
&= B^\mu(x,a).
\end{aligned}$$

■

From Lemma 4, we notice that $\delta_n\psi_n$ and $\epsilon_n\psi_n$ are unbiased estimates of $\nabla\rho(\mu)$ and $\nabla\eta(\mu)$, respectively, where $\psi_n = \psi(x_n,a_n) = \nabla\log\mu(a_n|x_n)$ is the *compatible* feature (see e.g., [65, 44]).

6 Average Reward Risk-Sensitive Actor-Critic Algorithm

We now present our risk-sensitive actor-critic algorithm for average reward MDPs. Algorithm 2 presents the complete structure of the algorithm along with the update rules for the average rewards $\widehat{\rho}_n, \widehat{\eta}_n$; TD errors δ_n, ϵ_n; critic v_n, u_n; and actor θ_n, λ_n parameters. The projection operators Γ and Γ_λ are as defined in Section 4, and similar to the discounted setting, are necessary for the convergence proof of the algorithm. The step-size schedules satisfy (A3) defined in Section 4, plus the step size schedule $\{\zeta_4(n)\}$ satisfies $\zeta_4(n) = k\zeta_3(n)$, for some positive constant k. This is to ensure that the average and critic updates are on the (same) fastest time-scale $\{\zeta_4(n)\}$ and $\{\zeta_3(n)\}$, the policy parameter update is on the intermediate time-scale $\{\zeta_2(n)\}$, and the Lagrange multiplier update is on the slowest time-scale $\{\zeta_1(n)\}$. This results in a three time-scale stochastic approximation algorithm.

Algorithm 2 Template of the Average Reward Risk-Sensitive Actor-Critic Algorithm

```
Input: parameterized policy $\mu(\cdot|\cdot;\theta)$ and value function feature vectors $\phi_v(\cdot)$ and $\phi_u(\cdot)$
Initialization: policy parameters $\theta = \theta_0$; value function weight vectors $v = v_0$ and $u = u_0$; initial state $x_0 \sim P_0(x)$
for $t = 0, 1, 2, \dots$ do
    Draw action $a_n \sim \mu(\cdot|x_n;\theta_n)$ and observe the next state $x_{n+1} \sim P(\cdot|x_n, a_n)$ and the reward $R(x_n, a_n)$

    Average Updates: $\widehat{\rho}_{n+1} = \big(1 - \zeta_4(n)\big)\widehat{\rho}_n + \zeta_4(n)R(x_n, a_n),$
                     $\widehat{\eta}_{n+1} = \big(1 - \zeta_4(n)\big)\widehat{\eta}_n + \zeta_4(n)R(x_n, a_n)^2$
    TD Errors: $\delta_n = R(x_n, a_n) - \widehat{\rho}_{n+1} + v_n^\top\phi_v(x_{n+1}) - v_n^\top\phi_v(x_n)$
               $\epsilon_n = R(x_n, a_n)^2 - \widehat{\eta}_{n+1} + u_n^\top\phi_u(x_{n+1}) - u_n^\top\phi_u(x_n)$
    Critic Update: $v_{n+1} = v_n + \zeta_3(n)\delta_n\phi_v(x_n), \qquad u_{n+1} = u_n + \zeta_3(n)\epsilon_n\phi_u(x_n)$    (46)
    Actor Update: $\theta_{n+1} = \Gamma\Big(\theta_n - \zeta_2(n)\big(-\delta_n\psi_n + \lambda_n(\epsilon_n\psi_n - 2\widehat{\rho}_{n+1}\delta_n\psi_n)\big)\Big)$    (47)
                  $\lambda_{n+1} = \Gamma_\lambda\Big(\lambda_n + \zeta_1(n)(\widehat{\eta}_{n+1} - \widehat{\rho}_{n+1}^2 - \alpha)\Big)$    (48)

end for
return policy and value function parameters $\theta, \lambda, v, u$
```

As in the discounted setting, the critic uses linear approximation for the differential value and square value functions, i.e., $\widehat{V}(x) = v^\top\phi_v(x)$ and $\widehat{U}(x) = u^\top\phi_u(x)$, where $\phi_v(\cdot)$ and $\phi_u(\cdot)$ are feature vectors of size κ_2 and κ_3, respectively. Although our estimates of $\rho(\theta)$ and $\eta(\theta)$ are unbiased, since we use biased estimates for V^θ and U^θ (linear approximations in the critic), our gradient estimates $\nabla\rho(\theta)$ and $\nabla\eta(\theta)$, and as a result $\nabla L(\theta, \lambda)$, are biased. The following lemma shows the bias in our estimate of $\nabla L(\theta, \lambda)$.

Lemma 5 *The bias of our actor-critic algorithm in estimating $\nabla L(\theta, \lambda)$ for fixed θ and λ is*

$$\mathcal{B}(\theta, \lambda) = \sum_x \boldsymbol{D}^\theta(x)\Big(-\big(1 + 2\lambda\rho(\theta)\big)\big[\nabla\bar{V}^\theta(x) - \nabla v^{\theta\top}\phi_v(x)\big] \\ + \lambda\big[\nabla\bar{U}^\theta(x) - \nabla u^{\theta\top}\phi_u(x)\big]\Big),$$

where $v^{\theta\top}\phi_v(\cdot)$ and $u^{\theta\top}\phi_u(\cdot)$ are estimates of $V^\theta(\cdot)$ and $U^\theta(\cdot)$ upon convergence of the TD recursion, and

$$\bar{V}^\theta(x) = \sum_a \mu(a|x)\big[r(x,a) - \rho(\theta) + \sum_{x'} P(x'|x,a) v^{\theta\top}\phi_v(x')\big],$$
$$\bar{U}^\theta(x) = \sum_a \mu(a|x)\big[r(x,a)^2 - \eta(\theta) + \sum_{x'} P(x'|x,a) u^{\theta\top}\phi_u(x')\big].$$

Proof The bias in estimating $\nabla L(\theta,\lambda)$ consists of the bias in estimating $\nabla\rho(\theta)$ and $\nabla\eta(\theta)$. Lemma 4 in Bhatnagar et al [14] shows the bias in estimating $\nabla\rho(\theta)$ as

$$\mathbb{E}[\delta_n^\theta \psi_n|\theta] = \nabla\rho(\theta) + \sum_{x\in\mathcal{X}} \boldsymbol{D}^\theta(x)\big[\nabla\bar{V}^\theta(x) - \nabla v^{\theta\top}\phi_v(x)\big],$$

where $\delta_n^\theta = R(x_n,a_n) - \widehat{\rho}_{n+1} + v^{\theta\top}\phi_v(x_{n+1}) - v^{\theta\top}\phi_v(x_n)$. Similarly we can prove that the bias in estimating $\nabla\eta(\theta)$ is

$$\mathbb{E}[\epsilon_n^\theta \psi_n|\theta] = \nabla\eta(\theta) + \sum_{x\in\mathcal{X}} \boldsymbol{D}^\theta(x)\big[\nabla\bar{U}^\theta(x) - \nabla u^{\theta\top}\phi_u(x)\big],$$

where $\epsilon_n^\theta = R(x_n,a_n) - \widehat{\eta}_{n+1} + u^{\theta\top}\phi_u(x_{n+1}) - u^{\theta\top}\phi_u(x_n)$. The claim follows by putting these two results together and given the fact that $\nabla\Lambda(\theta) = \nabla\eta(\theta) - 2\rho(\theta)\nabla\rho(\theta)$ and $\nabla L(\theta,\lambda) = -\nabla\rho(\theta) + \lambda\nabla\Lambda(\theta)$. Note that the following fact holds for the bias in estimating $\nabla\rho(\theta)$ and $\nabla\eta(\theta)$:

$$\sum_x \boldsymbol{D}^\theta(x)\big[\bar{V}^\theta(x) - v^{\theta\top}\phi_v(x)\big] = 0, \qquad \sum_x \boldsymbol{D}^\theta(x)\big[\bar{U}^\theta(x) - u^{\theta\top}\phi_u(x)\big] = 0.$$

■

Remark 7 (**Extension to Sharpe Ratio Optimization**)

The gradient of the Sharpe Ratio (SR) in the average setting is given by

$$\nabla S(\theta) = \frac{1}{\sqrt{\Lambda(\theta)}}\big(\nabla\rho(\theta) - \frac{\rho(\theta)}{2\Lambda(\theta)}\nabla\Lambda(\theta)\big),$$

and thus, the actor recursion for the SR-variant of our average reward risk-sensitive actor-critic algorithm is as follows:

$$\theta_{n+1} = \Gamma\Big(\theta_n + \frac{\zeta_2(n)}{\sqrt{\widehat{\eta}_{n+1} - \widehat{\rho}_{n+1}^2}}\big(\delta_n\psi_n - \frac{\widehat{\rho}_{n+1}(\epsilon_n\psi_n - 2\widehat{\rho}_{n+1}\delta_n\psi_n)}{2(\widehat{\eta}_{n+1} - \widehat{\rho}_{n+1}^2)}\big)\Big). \tag{49}$$

Note that the rest of the updates, including the average reward, TD errors, and critic recursions are as in the risk-sensitive actor-critic algorithm presented in Algorithm 2. Similar to the discounted setting, since there is no Lagrange multiplier in the SR optimization, the resulting actor-critic algorithm is a two time-scale stochastic approximation algorithm.

Remark 8 In the discounted setting, another popular variability measure is the *discounted normalized variance* [28]

$$\Lambda(\mu) = \mathbb{E}\left[\sum_{n=0}^\infty \gamma^n\big(R_n - \rho_\gamma(\mu)\big)^2\right], \tag{50}$$

where $\rho_\gamma(\mu) = \sum_{x,a} d^\mu_\gamma(x|x^0)\mu(a|x)r(x,a)$ and $d^\mu_\gamma(x|x^0)$ is the γ-discounted visiting distribution of state x under policy μ, defined in Section 2. The variability measure (50) has close resemblance to the average reward variability measure (39), and thus, any (discounted) risk measure based on (50) can be optimized similar to the corresponding average reward risk measure (39).

Remark 9 **(Simultaneous perturbation analogues)** In the average reward setting, a simultaneous perturbation algorithm would estimate the average reward ρ and the square reward η on the faster timescale and use these to estimate the gradient of the performance objective. However, a drawback with this approach, compared to the algorithm proposed above is the necessity for having two simulated trajectories (instead of one) for each policy update.

In the following section, we establish the convergence of our average reward actor-critic algorithm to a (local) saddle point of the risk-sensitive objective function $L(\theta, \lambda)$.

7 Convergence Analysis of the Discounted Reward Risk-Sensitive Actor-Critic Algorithms

Our proposed actor-critic algorithms use multi-timescale stochastic approximation and we use the ordinary differential equation (ODE) approach (see Chapter 6 of [21]) to analyze their convergence. We first provide the analysis for the SPSA based first-order algorithm RS-SPSA-G in Section 7.1 and later provide the necessary modifications to the proof of SF based first-order algorithm and SPSA/SF based second-order algorithms.

7.1 Convergence of the First-Order Algorithm: RS-SPSA-G

Recall that RS-SPSA-G is a two-loop scheme where the inner loop is a TD critic that evaluates the value/square value functions for both unperturbed as well as perturbed policy parameter. On the other hand, the outer loop is a two-timescale stochastic approximation algorithm, where the faster timescale updates policy parameter θ in the descent direction using SPSA estimates of the gradient of the Lagrangian and the slower timescale performs dual ascent for the Lagrange multiplier λ using sample constraint values. The faster timescale θ-recursion sees the λ-updates on the slower timescales as quasi-static, while the slower timescale λ-recursion sees the θ-updates as equilibrated.

The proof of convergence of the RS-SPSA-G algorithm to a (local) saddle point of the risk-sensitive objective function $\widehat{L}(\theta,\lambda) \triangleq -\widehat{V}^\theta(x^0) + \lambda(\widehat{\Lambda}^\theta(x^0) - \alpha) = -\widehat{V}^\theta(x^0) + \lambda\big(\widehat{U}^\theta(x^0) - \widehat{V}^\theta(x^0)^2 - \alpha\big)$ contains the following three main steps:

Step 1: Critic's Convergence. We establish that, for any given values of θ and λ that are updated on slower timescales, the TD critic converges to a fixed point of the projected Bellman operator for value and square value functions.

Step 2: Convergence of θ-recursion. We utilize the fact that owing to projection, the θ parameter is stable. Using a Lyapunov argument, we show that the θ-recursion tracks the ODE (55) in the asymptotic limit, for any given value of λ on the slowest timescale.

Step 3: Convergence of λ-recursion. This step is similar to earlier analysis for constrained MDPs . In particular, we show that λ-recursion in (19) converges and the overall convergence of (θ_n, λ_n) is to a local saddle point $(\theta^{\lambda^*}, \lambda^*)$ of $\widehat{L}(\theta,\lambda)$, with θ^{λ^*} satisfying the variance constraint in (3).

Step 1: (Critic's Convergence) Since the critic's update is in the inner loop, we can assume in this analysis that θ and λ are time-invariant quantities. The following theorem shows that the TD critic estimates for the value and square value function converge to the fixed point given by (12), for any given policy θ.

Theorem 2 *Under (A1)-(A4), for any given policy parameter θ and Lagrange multiplier λ, the critic parameters $\{v_m\}$ and $\{u_m\}$ governed by the recursions of* (13) *converge almost surely, i.e.,*

$$\textit{As } m \to \infty, v_m \to \bar{v} \textit{ and } u_m \to \bar{u} \textit{ a.s.}$$

In the above $\bar{v}$ and $\bar{u}$ are the solutions to the TD fixed point equations for policy θ (see (12) *in Section 4.2.*

Remark 10 It is easy to conclude from the above theorem that the TD critic parameters for the perturbed policy parameter also converge almost surely, i.e., $v_m^+ \to \bar{v}^+$ and $u_m^+ \to \bar{u}^+$ a.s., where $\bar{v}^+$ and $\bar{u}^+$ are the unique solutions to TD fixed point relations for perturbed policy $\theta_n + \beta_n \Delta_n$, where θ_n, β_n and Δ_n correspond to the policy parameter, perturbation constant and perturbation random variable. The latter quantities are updated in the outer loop - see Algorithm 1.

We first provide a proof of Lemma 2 (see Section 4.2), which claimed that the operator ΠT for the value/square value functions is a contraction mapping. The result in Lemma 2 is essential in establishing the convergence result in Theorem 2.

Proof **(Lemma 2)** We employ the technique from [69] to prove this result. First, it is well-known that $\Pi_v T_v^\theta$ is a contraction mapping (cf. Lemma 6 in [71]). This can be inferred as follows: For any $y, \bar{y} \in \mathbb{R}^{2|\mathcal{X}|}$,

$$\|T_v^\theta y - T_v^\theta \bar{y}\|_{\boldsymbol{D}^\theta} = \gamma \|y_v - \bar{y}_v\|_{\boldsymbol{D}^\theta}.$$

We have used the fact that $\|P^\theta v\|_{\boldsymbol{D}^\theta} \leq \|v\|_{\boldsymbol{D}^\theta}$ for any $v \in \mathbb{R}^{|\mathcal{X}|}$ (For a proof, see Lemma 1 in [71]). The claim that $\Pi_v T_v^\theta$ is a contraction mapping now follows from the fact that the projection operator Π_v is non-expansive under $\|\cdot\|_{\boldsymbol{D}^\theta}$ norm.

Now, for any $y, \bar{y} \in \mathbb{R}^{2|\mathcal{X}|}$, we have

$$\begin{aligned}
&\|\Pi_u T_u^\theta y - \Pi_u T_u^\theta \bar{y}\|_{\boldsymbol{D}^\theta} \\
=&\|2\gamma \Pi_u R^\theta P^\theta y_v - 2\gamma \Pi_u R^\theta P^\theta \bar{y}_v + \gamma^2 \Pi_u P^\theta y_u - \gamma^2 \Pi_u P^\theta \bar{y}_u\|_{\boldsymbol{D}^\theta} \\
\leq&2\gamma \|\Pi_u R^\theta P^\theta y_v - \Pi_u R^\theta P^\theta \bar{y}_v\|_{\boldsymbol{D}^\theta} + \gamma^2 \|y_u - \bar{y}_u\|_{\boldsymbol{D}^\theta} \\
\leq&\gamma C_1 \|y_v - \bar{y}_v\|_{\boldsymbol{D}^\theta} + \gamma^2 \|y_u - \bar{y}_u\|_{\boldsymbol{D}^\theta}, \qquad (51)
\end{aligned}$$

for some $C_1 < \infty$. The first inequality above follows from the aforementioned facts that P^θ and Π_u are non-expansive. The second inequality follows by using equivalence of norms (cf. the justification for Eq. (7) in the proof of Lemma 7 in [70]).

Setting $\nu = \dfrac{\gamma C_1}{\epsilon + \gamma C_1}$, where ϵ is such that $\gamma + \epsilon < 1$ and plugging in (51), we obtain

$$\begin{aligned}
&\|\Pi T^\theta y - \Pi T^\theta \bar{y}\|_\nu \\
=&\nu \|T_v^\theta y - T_v^\theta \bar{y}\|_{\boldsymbol{D}^\theta} + (1-\nu)\|\Pi_u T_u^\theta y - \Pi_u T_u^\theta \bar{y}\|_{\boldsymbol{D}^\theta} \\
\leq&\nu\gamma \|y_v - \bar{y}_v\|_{\boldsymbol{D}^\theta} + (1-\nu)\gamma C_1 \|y_v - \bar{y}_v\|_{\boldsymbol{D}^\theta} + (1-\nu)\gamma^2 \|y_u - \bar{y}_u\|_{\boldsymbol{D}^\theta} \\
\leq&\nu(\gamma+\epsilon)\|y_v - \bar{y}_v\|_{\boldsymbol{D}^\theta} + (1-\nu)\gamma \|y_u - \bar{y}_u\|_{\boldsymbol{D}^\theta} \\
\leq&(\gamma+\epsilon)\|y - \bar{y}\|_\nu.
\end{aligned}$$

The claim follows by setting $\bar{\gamma} = \gamma + \epsilon$. ■

Proof **(Theorem 2)** The v-recursion in (13) is performing TD) with function approximation for the value function, while the u-recursion is doing the same for the square value function. The convergence of v-recursion to the fixed point in (12) can be inferred from [71].

Using an approach similar to [69], we club both v and u recursions and establish convergence using a stability argument in the following: Let $w_m = (v_m, u_m)^\intercal$. Then, (13) can be seen to be equivalent to

$$\begin{aligned} w_{m+1} =& w_m + \zeta_3(m)(Mw_m + \xi + \varDelta M_{m+1}), \text{ where} \quad (52)\\ M =& \begin{pmatrix} \varPhi_v^\intercal \boldsymbol{D}^\theta(\gamma P^\theta - I)\varPhi_v & 0 \\ 2\gamma \varPhi_u^\intercal \boldsymbol{D}^\theta R^\theta P^\theta \varPhi_v & \varPhi_u^\intercal \boldsymbol{D}^\theta(\gamma^2 P^\theta - I)\varPhi_u \end{pmatrix} \text{ and}\\ \xi =& \begin{pmatrix} \varPhi_v^\intercal \boldsymbol{D}^\theta r^\theta \\ \varPhi_u^\intercal \boldsymbol{D}^\theta R^\theta r^\theta \end{pmatrix}. \end{aligned}$$

Further, $\varDelta M_{m+1}$ is a martingale difference, i.e., $\mathbb{E}[\varDelta M_{m+1} \mid \mathcal{F}_m] = 0$, where $\mathcal{F}_m$ is the sigma field generated by $w_l, \varDelta M_l, l \leq m.$

Let $h(w) = Mw + \xi$. Then, the ODE associated with (52) is

$$\dot{w}_t = h(w_t). \quad (53)$$

The above ODE has a unique globally asymptotically stable equilibrium, since M is a negative definite. To see the latter fact, observe that M is block triangular and hence its eigenvalues are that of $\varPhi_v^\intercal \boldsymbol{D}^\theta(\gamma P^\theta - I)\varPhi_v$ and $\varPhi_u^\intercal \boldsymbol{D}^\theta(\gamma^2 P^\theta - I)\varPhi_u$. It can be inferred from Theorem 2 of [71] that the aforementioned matrices are negative definite. For the sake of completeness, we provide a brief sketch in the following: For any $V \in \mathbb{R}^{|\mathcal{X}|}$, it can be shown that $\left\|P^\theta V\right\|_{\boldsymbol{D}^\theta} \leq \|V\|_{\boldsymbol{D}^\theta}$ (see Lemma 1 in [71] for a proof). Now,

$$\begin{aligned} V^\intercal \boldsymbol{D}^\theta \gamma P^\theta V \leq& \gamma \left\|(\boldsymbol{D}^\theta)^{1/2} V\right\| \left\|(\boldsymbol{D}^\theta)^{1/2} PV\right\| \\ =& \gamma \|V\|_{\boldsymbol{D}^\theta} \|PV\|_{\boldsymbol{D}^\theta} \\ \leq& \gamma \|V\|^2_{\boldsymbol{D}^\theta}. \end{aligned}$$

Hence, $V^\intercal \boldsymbol{D}^\theta(\gamma P^\theta - I)V \leq (\gamma - 1)\|V\|^2_{\boldsymbol{D}^\theta} < 0$. By (A3), we know that $\varPhi_v$ is full rank implying the negative definiteness of $\varPhi_v^\intercal \boldsymbol{D}^\theta(\gamma P^\theta - I)\varPhi_v$. Using the same argument as above and replacing $\varPhi_v$ with $\varPhi_u$ and γ with γ^2, one can conclude that $\varPhi_u^\intercal \boldsymbol{D}^\theta(\gamma^2 P^\theta - I)\varPhi_u$.

The final claim now follows by applying Theorems 2.1-2.2(i) of [23], provided we verify assumptions (A1)-(A2) there. The latter assumptions are given as follows:
(A1) The function h is Lipschitz. For any c, define $h_c(w) = h(cw)/c$. Then, there exists a continuous function h_∞ such that $h_c \to h_\infty$ as $c \to \infty$ uniformly on compacts. Furthermore, origin is an asymptotically stable equilibrium for the ODE

$$\dot{w}_t = h_\infty(w_t). \quad (54)$$

(A2) The martingale difference $\{\varDelta M_m, m \geq 1\}$ is square-integrable with

$$\mathbb{E}[\|\varDelta M_{m+1}\|^2 \mid \mathcal{F}_m] \leq C_0(1 + \|w_m\|^2), m \geq 0,$$

where $C_0 < \infty$.

It is straightforward to verify (A1), as $h_c(w) = Mw + \xi/c$ converges to $h_\infty(w) = Mw$ as $c \to \infty$. Given that M is negative definite, it is easy to see that origin is a asymptotically stable equilibrium for the ODE (54). (A2) can also be verified by using the same arguments that were used to show that the martingale difference associated with the regular TD algorithm with function approximation satisfies a bound on the second moment (cf. [71]). ■

Step 2: (Analysis of θ-recursion) Due to timescale separation, the value of λ (updated on a slower timescale) is assumed to be constant for the analysis of the θ-update. To see this in rigorous terms, first rewrite the λ-recursion as

$$\lambda_{n+1} = \Gamma_\lambda \left[\lambda_n + \zeta_2(n) \hat{H}(n) \right].$$

where $\hat{H}(n) = \frac{\zeta_1(n)}{\zeta_2(n)} \left(u_n^\mathsf{T} \phi_u(x^0) - \left(v_n^\mathsf{T} \phi_v(x^0)\right)^2 - \alpha \right)$. Since the critic recursions converge, it is easy to see that $\sup_n \hat{H}(n)$ is finite. Combining with the observation that $\frac{\zeta_1(n)}{\zeta_2(n)} = o(1)$ due to the assumption (A3) on step-sizes, we see that the λ-recursion above tracks the ODE $\dot{\lambda} = 0$.

In the following, we show that the update of θ is equivalent to gradient descent for the function $\widehat{L}(\theta, \lambda)$ and converges to a limiting set that depends on λ.

Consider the following ODE

$$\dot{\theta}_t = \check{\Gamma} \left(\nabla \widehat{L}(\theta_t, \lambda) \right), \tag{55}$$

with the limiting set $\mathcal{Z}_\lambda = \left\{ \theta \in C : \check{\Gamma}\left(\nabla \widehat{L}(\theta_t, \lambda)\right) = 0 \right\}$. In the above, $\check{\Gamma}(\cdot)$ is a projection operator that ensures the evolution of θ via the ODE (55) stays within the set $\Theta := \prod_{i=1}^{\kappa_1} [\theta_{\min}^{(i)}, \theta_{\max}^{(i)}]$ and is defined as follows: For any bounded continuous function $f(\cdot)$,

$$\check{\Gamma}\left(f(\theta)\right) = \lim_{\tau \to 0} \frac{\Gamma\left(\theta + \tau f(\theta)\right) - \theta}{\tau}. \tag{56}$$

Notice that the limit above may not exist and in that case, as pointed out on pp. 191 of [36], one can define $\check{\Gamma}(f(\theta))$ to be the set of all possible limit points. From the definition above, it can be inferred that for θ in the interior of Θ, $\check{\Gamma}(f(\theta)) = f(\theta)$, while for θ on the boundary of Θ, $\check{\Gamma}(f(\theta))$ is the projection of $f(\theta)$ onto the tangent space of the boundary of Θ at θ.

The main result regarding the convergence of the policy parameter θ for both the RS-SPSA-G and RS-SF-G algorithms is as follows:

Theorem 3 *Under (A1)-(A4), for any given Lagrange multiplier λ, θ_n updated according to* (19) *converges almost surely to $\theta^* \in \mathcal{Z}_\lambda$.*

The proof of the above theorem requires the following lemma which shows that the conditions m_n, β_n in (A4) ensure that the TD-critic does not introduce any bias from a finite sample run length of m_n.

Lemma 6 *Let*

$$\begin{aligned} \mathcal{T}_n^{(i)} &\triangleq \left(\left(1 + 2\lambda v_n^\mathsf{T} \phi_v(x^0)\right) \frac{(v_n^+ - v_n)^\mathsf{T} \phi_v(x^0)}{\beta_n \Delta_n^{(i)}} - \lambda \frac{(u_n^+ - u_n)^\mathsf{T} \phi_u(x^0)}{\beta_n \Delta_n^{(i)}} \right), \\ \widehat{L}(\theta, \lambda) &\triangleq - \widehat{V}^\theta(x^0) + \lambda\left(\widehat{U}^\theta(x^0) - \widehat{V}^\theta(x^0)^2 - \alpha\right), \end{aligned}$$

where $\widehat{V}(\theta) = \phi_{\bar{v}}(x^0)^\mathsf{T} \bar{v}$ and $\widehat{U}(\theta) = \phi_{\bar{u}}(x^0)^\mathsf{T} \bar{u}$ denote the approximate value and square value functions for policy θ[7].

Then, we have that

$$\left| \mathbb{E}\left(\mathcal{T}_n^{(i)} \mid \theta_n \right) - \nabla \widehat{L}(\theta_n, \lambda) \right| = O(\beta_n^2), \textit{ for } i = 1, \ldots, \kappa_1.$$

[7] For notational convenience, we drop the dependence of $\bar{v}$ and $\bar{u}$ on the underlying policy parameter θ and this dependence should be clear from the context.

Proof Let

$$\xi_{1,n} := \bigg(\mathcal{T}_n^{(i)} - \bigg(\big(1 + 2\lambda \bar{v}^\mathsf{T} \phi_v(x^0)\big) \frac{(\bar{v}^+ - \bar{v})^\mathsf{T} \phi_v(x^0)}{\beta_n \Delta_n^{(i)}} - \lambda \frac{(\bar{u}^+ - \bar{u})^\mathsf{T} \phi_u(x^0)}{\beta_n \Delta_n^{(i)}} \bigg).$$

From Theorem 1, we know that the critic parameters v_n, u_n converge to their limits $\bar{v}, \bar{u}$ at the rate $O(m^{-1/2})$ and hence, after m_n steps of the TD-critic, $\xi_{1,n} = O(\frac{1}{\sqrt{m_n}\beta_n})$. Now, from (A4), we have that $\frac{1}{\sqrt{m_n}\beta_n} \to 0$ and hence $\xi_{1,n}$ vanishes asymptotically. Hence, we have

$$\mathcal{T}_n^{(i)} \to \bigg(\big(1 + 2\lambda \bar{v}^\mathsf{T} \phi_v(x^0)\big) \frac{(\bar{v}^+ - \bar{v})^\mathsf{T} \phi_v(x^0)}{\beta \Delta_n^{(i)}} - \lambda \frac{(\bar{u}^+ - \bar{u})^\mathsf{T} \phi_u(x^0)}{\beta \Delta_n^{(i)}} \bigg) \bigg). \tag{57}$$

We next show that the RHS above is an order $O(\beta_n^2)$ term away from the gradient of the Lagrangian $L(\theta_n, \lambda)$. Using a Taylor's expansion of $\widehat{V}(\cdot)$ around θ_n, we obtain:

$$\widehat{V}(\theta_n + \beta_n \Delta_n) = \widehat{V}(\theta_n) + \beta_n {\Delta_n}^\mathsf{T} \nabla \widehat{V}(\theta_n) + \frac{{\beta_n}^2}{2} {\Delta_n}^\mathsf{T} \nabla^2 \widehat{V}(\theta_n) \Delta_n + O(\beta_n^3).$$

Taking expectations and rearranging terms, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{\widehat{V}(\theta_n + \beta_n \Delta_n) - \widehat{V}(\theta_n)}{\beta_n \Delta_n^{(i)}} \right) \middle| \, \theta_n \right] \\
=&\mathbb{E}\left[\frac{\Delta_n^\mathsf{T} \nabla \widehat{V}(\theta_n)}{\Delta_n^{(i)}} \mid \theta_n \right] + \mathbb{E}\left[\frac{\Delta_n^\mathsf{T} \nabla_{\theta_n}^2 \widehat{V}(\theta_n) \Delta_n}{\Delta_n^{(i)}} \mid \theta_n \right] + O(\beta_n^2) \\
=&\nabla_i \widehat{V}(\theta_n) + \mathbb{E}\left[\sum_{j \neq i} \frac{\Delta_n^{(j)}}{\Delta_n^{(i)}} \nabla_j \widehat{V}(\theta_n) \mid \theta_n \right] + O(\beta_n^2) \\
=&\nabla_i \widehat{V}(\theta_n) + O(\beta_n^2).
\end{aligned} \tag{58}$$

In the above, we have used the fact that Δ_n is i.i.d. Rademacher and independent of θ_n.

In a similar manner, defining $\widehat{U}(\theta_n) = \phi_{\bar{u}}(x^0)^\mathsf{T} \bar{u}$ and $\widehat{U}(\theta_n + \beta_n \Delta_n) = \phi_{\bar{u}^+}(x^0)^\mathsf{T} \bar{u}^+$, we can conclude that

$$\mathbb{E}\left[\left(\frac{\widehat{U}(\theta_n + \beta_n \Delta_n) - \widehat{U}(\theta_n)}{\beta_n {\Delta_n}^{(i)}} \right) \middle| \, \theta_n \right] = \nabla_i \widehat{U}(\theta_n) + O(\beta_n^2). \tag{59}$$

The claim now follows by plugging in (58)–(59) into (57).

■

In order to the prove Theorem 3, we require the well-known Kushner-Clark lemma (see [36, pp. 191-196]). For the sake of completeness, we recall this result below.

Kushner-Clark lemma. Consider the following recursion in κ_1-dimensions:

$$x_{n+1} = \Gamma(x_n + a(n)(h(x_n) + \xi_{1,n} + \xi_{2,n})), \tag{60}$$

where Γ projects the iterate x_n onto a compact and convex set, say $C \in \mathbb{R}^N$. The ODE associated with (60) is given by

$$\dot{x}(t) = \bar{\Gamma}(h(x(t))), \tag{61}$$

where $\bar{\Gamma}$ is a projection operator that keeps the ODE evolution within the set C and is defined as in (56).

We make the following assumptions:

(B1) h is a continuous $\mathbb{R}^{\kappa_1}$-valued function.
(B2) The sequence $\xi_{1,n}, n \geq 0$ is a bounded random sequence with $\xi_{1,n} \to 0$ almost surely as $n \to \infty$.
(B3) The step-sizes $a(n), n \geq 0$ satisfy $a(n) \to 0$ as $n \to \infty$ and $\sum_n a(n) = \infty$.
(B4) $\{\xi_{2,n}, n \geq 0\}$ is a sequence such that for any $\epsilon > 0$,

$$\lim_{n\to\infty} P\left(\sup_{m\geq n}\left\|\sum_{i=n}^{m} a_i \xi_{1,i}\right\| \geq \epsilon\right) = 0.$$

(B5) The ODE (61) has a compact subset K of $\mathcal{R}^{\kappa_1}$ as its set of asymptotically stable equilibrium points.

The main result (see [36, pp. 191-196]) is as follows:

Theorem 4 *Assume (B1)–(B5). Then, x_n converges almost surely to the set K.*

Proof (**Theorem 3**) We first rewrite the recursion (19) as follows:

$$\theta^{(i)}_{n+1} = \Gamma_i\bigg(\theta^{(i)}_n + \zeta_2(n)\Big(\nabla \widehat{L}(\theta_n, \lambda) + \xi_{1,n} + \xi_{2,n}\Big)\bigg), \tag{62}$$

where

$$\begin{aligned}\xi_{1,n} =& \mathbb{E}\left(\mathcal{T}^{(i)}_n \mid \theta_n\right) - \nabla \widehat{L}(\theta_n, \lambda),\\ \xi_{2,n} =& \mathcal{T}^{(i)}_n - \mathbb{E}\left(\mathcal{T}^{(i)}_n \mid \theta_n\right),\end{aligned}$$

with $\mathcal{T}^{(i)}_n$ defined as in Lemma 6.

We now verify (B1)- (B5) for the above recursion:

- From (A1) together with the facts that the state space is finite and the projection Γ is onto a compact set, we have from Theorem 2 of [53] that the stationary distributions $\boldsymbol{D}^{\theta}_{\gamma}(x|x^0)$ and $\widetilde{d}^{\theta}_{\gamma}(x|x^0)$ are continuously differentiable. This in turn implies continuity of $\nabla \widehat{V}(\theta_n)$ and $\nabla \widehat{U}(\theta_n)$. Thus, (B1) follows for $\nabla \widehat{L}(\theta_n, \lambda)$.
- In light of Lemma 6 and (A4), we have that $\xi_{1,n} \to 0$ as $n \to \infty$.
- (A4) implies (B3).

- A simple calculation shows that $\mathbb{E}(\xi_{2,n})^2 \leq \mathbb{E}(\mathcal{T}_n^{(i)})^2 \leq C_3/\beta_n^2$ for some $C_3 < \infty$. Applying Doob's inequality, we obtain

$$P\left(\sup_{l \geq k} \left\| \sum_{n=k}^{l} \zeta_2(n)\xi_{2,n} \right\| \geq \epsilon \right) \leq \frac{1}{\epsilon^2} \sum_{n=k}^{\infty} \zeta_2(n)^2 \mathbb{E} \left\| \xi_{2,n} \right\|^2 . \tag{63}$$

$$\leq \frac{C_3}{\epsilon^2} \sum_{n=k}^{\infty} \frac{\zeta_2(n)^2}{\beta_n^2} = 0. \tag{64}$$

  Thus, (B4) is satisfied.
- $\mathcal{Z}_\lambda$ is an asymptotically stable attractor for the ODE (55), with $\widehat{L}(\theta,\lambda)$ itself serving as a strict Lyapunov function. This can be inferred as follows:

$$\frac{d\widehat{L}(\theta,\lambda)}{dt} = \nabla \widehat{L}(\theta,\lambda)\dot{\theta} = \nabla \widehat{L}(\theta,\lambda)\check{\varGamma}\big(-\nabla \widehat{L}(\theta,\lambda)\big) < 0.$$

The claim now follows from Kushner-Clark lemma. ■

Step 3: (Analysis of λ-recursion and Convergence to a Local Saddle Point) We first show that the λ-recursion converges and then prove that the whole algorithm converges to a local saddle point of $\widehat{L}(\theta,\lambda)$.

We define the following ODE governing the evolution of λ:

$$\dot{\lambda}_t \;=\; \check{\varGamma}_\lambda\big[\widehat{\varLambda}^{\theta^{\lambda_t}}(x^0) - \alpha\big] \;=\; \check{\varGamma}_\lambda\big[\widehat{U}^{\theta^{\lambda_t}}(x^0) - \widehat{V}^{\theta^{\lambda_t}}(x^0)^2 - \alpha\big], \tag{65}$$

where θ^{λ_t} is the limiting point of the θ-recursion corresponding to λ_t. Further, $\check{\varGamma}_\lambda$ is an operator similar to the operator $\check{\varGamma}$ defined in (56) and is defined as follows: For any bounded continuous function $f(\cdot)$,

$$\check{\varGamma}_\lambda\big(f(\lambda)\big) = \lim_{\tau \to 0} \frac{\varGamma_\lambda\big(\lambda + \tau f(\lambda)\big) - \lambda}{\tau}. \tag{66}$$

Theorem 5 *$\lambda_n \to \mathcal{F}$ almost surely as $n \to \infty$, where $\mathcal{F} \overset{\triangle}{=} \big\{\lambda \mid \lambda \in [0,\lambda_{\max}],\ \check{\varGamma}_\lambda\big[\widehat{\varLambda}^{\theta^\lambda}(x^0) - \alpha\big] = 0,\ \theta^\lambda \in \mathcal{Z}_\lambda\big\}$.*

Proof The proof follows using standard stochastic approximation arguments. The first step is to rewrite the λ-recursion as follows:

$$\lambda_{n+1} = \varGamma_\lambda\left[\lambda_n + \zeta_1(n)\Big(\bar{u}^\mathsf{T}\phi_u(x^0) - \big(\bar{v}^\mathsf{T}\phi_v(x^0)\big)^2 - \alpha + \xi_{2,n}\Big)\right],$$

where $\xi_{2,n} := \Big(u_n^\mathsf{T}\phi_u(x^0) - \big(v_n^\mathsf{T}\phi_v(x^0)\big)^2\Big) - \Big(\bar{u}^\mathsf{T}\phi_u(x^0) - \big(\bar{v}^\mathsf{T}\phi_v(x^0)\big)^2\Big)$. Note that the converged critic parameters $\bar{v}$ and $\bar{u}$ are for the policy θ^{λ_n}. The latter is a limiting point of the θ-recursion, with the Lagrange multiplier λ_n. Owing to convergence of θ-recursion and also TD-critic in the inner loop, we can conclude that $\xi_{2,n} = o(1)$. Thus, $\xi_{2,n}$ adds an asymptotically vanishing bias term to the λ-recursion above. The claim follows by applying the standard result in Theorem 2 of [21] for convergence of stochastic approximation schemes. ■

Recall that $\widehat{L}(\theta,\lambda) \stackrel{\triangle}{=} -\widehat{V}^\theta(x^0)+\lambda(\widehat{\Lambda}^\theta(x^0)-\alpha)$ and hence $\nabla_\lambda \widehat{L}(\theta,\lambda) = \widehat{\Lambda}^\theta(x^0)-\alpha$. Thus,

$$\check{\Gamma}_\lambda\big[\widehat{\Lambda}^{\theta^\lambda}(x^0)-\alpha\big]=0,$$

is the same as

$$\check{\Gamma}_\lambda \nabla_\lambda \widehat{L}(\theta^\lambda,\lambda)=0.$$

As in [20], we invoke the envelope theorem of mathematical economics [40] to conclude that the ODE (65) is equivalent to the following

$$\dot{\lambda}_t = \check{\Gamma}_\lambda\big[\nabla_\lambda \widehat{L}(\theta^{\lambda_t},\lambda_t)\big]. \tag{67}$$

Note that the above has to interpreted in the *Cartheodory* sense, i.e., as the following integral equation

$$\lambda_t = \lambda_0 + \int_0^t \check{\Gamma}_\lambda\big[\nabla_\lambda \widehat{L}(\theta^{\lambda_s},\lambda_s)\big]ds.$$

As noted in Lemma 4.3 of [20], using the generalized envelope theorem from [42] it can be shown that the RHS of (67) coincides with that of (65) at differentiable points, while the ODE spends zero time at non-differentiable points (except at the points of maxima).

We next claim that the limit θ^{λ^*} corresponding to λ^* satisfies the variance constraint in (3), i.e.,

Proposition 1 *For any λ^* in $\hat{\mathcal{F}} \stackrel{\triangle}{=} \big\{\lambda \mid \lambda\in[0,\lambda_{\max}),\ \check{\Gamma}_\lambda\big[\widehat{\Lambda}^{\theta^\lambda}(x^0)-\alpha\big]=0,\ \theta^\lambda\in\mathcal{Z}_\lambda\big\}$, the corresponding limiting point θ^{λ^*} satisfies the variance constraint $\widehat{\Lambda}^{\theta^{\lambda^*}}(x^0)\le\alpha$.*

Proof Follows in a similar manner as Proposition 10.6 in [17].

From Theorems 3–5 and Proposition 1, it is evident that the actor recursion (19) converges to a tuple $(\theta^{\lambda^*},\lambda^*)$ that is a local minimum w.r.t. θ and a local maximum w.r.t. λ of $\widehat{L}(\theta,\lambda)$. In other words, overall convergence is to a (local) saddle point of $\widehat{L}(\theta,\lambda)$. Further, the limit is also feasible for the constrained problem in (3) as θ^{λ^*} satisfies the variance constraint there.

7.2 Convergence of the First-Order Algorithm: RS-SF-G

Note that since RS-SPSA-G and RS-SF-G use different methods to estimate the gradient, their proofs only differ in the second step, i.e., the convergence of the policy parameter θ.

Proof of Theorem 3 for SF

Proof As in the case of the SPSA algorithm, we rewrite the θ-update in (20) using the converged TD-parameters and constant λ as

$$\theta_{n+1}^{(i)} = \Gamma_i\bigg(\theta_n^{(i)} - \zeta_2(n)\Big(\frac{-\Delta_n^{(i)}\big(1+2\lambda\bar{v}^\intercal\phi_v(x^0)\big)}{\beta}(\bar{v}^+-\bar{v})^\intercal\phi_v(x^0) \\ + \frac{\lambda\Delta_n^{(i)}}{\beta}(\bar{u}^+-\bar{u})^\intercal\phi_u(x^0)+\xi_{1,n}\Big)\bigg),$$

where $\xi_{1,n} \to 0$ by using arguments analogous to those in the proof of Lemma 6. Next, we establish that $\mathbb{E}\left[\dfrac{\Delta^{(i)}}{\beta_n}(\bar{v}^+ - \bar{v})^\intercal \phi_v(x^0) \mid \theta, \lambda\right]$ is an asymptotically correct estimate of the gradient of $\widehat{V}(\theta)$ in the following:

$$\mathbb{E}\left[\frac{\Delta_n^{(i)}}{\beta_n}(\bar{v}^+ - \bar{v})^\intercal \phi_v(x^0) \mid \theta_n, \lambda\right] \longrightarrow \nabla_i \bar{v}^\intercal \phi_v(x^0) \text{ a.s. as } n \to \infty.$$

The above follows in a similar manner as Proposition 10.2 of Bhatnagar et al [17]. On similar lines, one can see that

$$\mathbb{E}\left[\frac{\Delta_n^{(i)}}{\beta_n}(\bar{u}^+ - \bar{u})^\intercal \phi_u(x^0) \mid \theta_n, \lambda\right] \longrightarrow \nabla_i \bar{u}^\intercal \phi_u(x^0) \text{ a.s. as } n \to \infty.$$

Thus, (20) can be seen to be a discretization of the ODE (55) and the rest of the analysis follows in a similar manner as in the SPSA proof. ■

7.2.1 Convergence of the Second-Order Algorithms: RS-SPSA-N and RS-SF-N

Convergence analysis of the second-order algorithms involves the same steps as that of the first-order algorithms. In particular, the first step involving the TD-critic and the third step involving the analysis of λ-recursion follow along similar lines as earlier, whereas θ-recursion analysis in the second step differs significantly.

Step 2: (Analysis of θ-recursion for RS-SPSA-N and RS-SF-N) Since the policy parameter is updated in the descent direction with a Newton decrement, the limiting ODE of the θ-recursion for the second order algorithms is given by

$$\dot{\theta}_t = \check{\Gamma}\left(\Upsilon\big(\nabla^2 L(\theta_t, \lambda)\big)^{-1} \nabla L(\theta_t, \lambda)\right), \tag{68}$$

where $\check{\Gamma}$ is as before (see (56)). Let

$$\mathcal{Z}_\lambda = \left\{\theta \in \Theta : -\nabla L(\theta_t, \lambda)^T \Upsilon\big(\nabla_\theta^2 L(\theta_t, \lambda)\big)^{-1} \nabla L(\theta_t, \lambda) = 0\right\}.$$

denote the set of asymptotically stable equilibrium points of the ODE (68) and $\mathcal{Z}_\lambda^\varepsilon$ its ε-neighborhood. Then, we have the following analogue of Theorem 3 for the RS-SPSA-N and RS-SF-N algorithms:

Theorem 6 *Under (A1)-(A5), for any given Lagrange multiplier λ and $\varepsilon > 0$, there exists $\beta_0 > 0$ such that for all $\beta \in (0, \beta_0)$, $\theta_n \to \theta^* \in \mathcal{Z}_\lambda^\varepsilon$ almost surely.*

Proof of Theorem 6 for RS-SPSA-N

Before we prove Theorem 6, we establish that the Hessian estimate H_n in (30) converges almost surely to the true Hessian $\nabla_\theta^2 L(\theta_n, \lambda)$ in the following lemma.

Lemma 7 *For all $i, j \in \{1, \ldots, \kappa_1\}$, we have the following claims with probability one:*

(i) $$\left\| \frac{L(\theta_n + \beta_n \Delta_n + \beta_n \widehat{\Delta}_n, \lambda) - L(\theta_n, \lambda)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} - \nabla^2_{\theta_n^{(i,j)}} L(\theta_n, \lambda) \right\| \to 0,$$

(ii) $$\left\| \frac{L(\theta_n + \beta_n \Delta_n + \beta_n \widehat{\Delta}_n, \lambda) - L(\theta_n, \lambda)}{\beta_n \widehat{\Delta}_n^{(i)}} - \nabla_{\theta_n^{(i)}} L(\theta_n, \lambda) \right\| \to 0,$$

(iii) $$\left\| H^{(i,j)} - \nabla^2_{\theta_n^{(i,j)}} L(\theta_n, \lambda) \right\| \to 0,$$

(iv) $$\left\| M - \Upsilon(\nabla^2_{\theta_n} L(\theta_n, \lambda))^{-1} \right\| \to 0.$$

Proof The proofs of the above claims follow from Propositions 10.10, 10.11 and Lemmas 7.10 and 7.11 of [17], respectively. ■

Proof **(Theorem 6 for RS-SPSA-N)** As in the case of the first order methods, due to timescale separation, we can treat $\lambda_n \equiv \lambda$, a constant and use the converged TD-parameters to arrive at the following equivalent update rules for the Hessian recursion (30) and θ-recursion (31):

$$\begin{aligned} H_{n+1}^{(i,j)} = H_n^{(i,j)} + \zeta_2'(n) \Bigg[& \frac{\big(1 + \lambda_n (\bar{v}_n + \bar{v}_n^+)^\intercal \phi_v(x^0)\big)(\bar{v}_n - \bar{v}_n^+)^\intercal \phi_v(x^0)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} \\ & + \frac{\lambda(\bar{u}_n^+ - \bar{u}_n)^\intercal \phi_u(x^0)}{\beta_n^2 \Delta_n^{(i)} \widehat{\Delta}_n^{(j)}} - H_n^{(i,j)} \Bigg], \\ \theta_{n+1}^{(i)} = \Gamma_i \Bigg[\theta_n^{(i)} + \zeta_2(n) \sum_{j=1}^{\kappa_1} M_n^{(i,j)} \Bigg(& \frac{\big(1 + 2\lambda \bar{v}_n^\intercal \phi_v(x^0)\big)(\bar{v}_n^+ - \bar{v}_n)^\intercal \phi_v(x^0)}{\beta_n \Delta_n^{(j)}} \\ & - \frac{\lambda(\bar{u}_n^+ - \bar{u}_n)^\intercal \phi_u(x^0)}{\beta_n \Delta_n^{(j)}} \Bigg) \Bigg]. \end{aligned}$$

By a completely parallel argument to the proof of Lemma 6 in conjunction with Lemma 7, the θ-recursion above is equivalent to the following:

$$\theta_{n+1}^{(i)} = \bar{\Gamma}_i \Big(\theta_n^{(i)} + \zeta_2(n) \big(\nabla^2 L(\theta_n, \lambda)\big)^{-1} \nabla L(\theta_n, \lambda) \Big). \tag{69}$$

The above can be seen as a discretization of the ODE (68), with $\mathcal{Z}_\lambda$ serving as its asymptotically stable attractor. The rest of the claim follows in a similar manner as Theorem 3. ■

Proof of Theorem 6 for RS-SF-N

Proof We first establish the following result for the gradient and Hessian estimators employed in RS-SF-N:

Lemma 8 *We have the following claims with probability one:*

(i) $$\left\| E\left[\tfrac{1}{\beta_n^2} \bar{H}(\Delta_n)(L(\theta_n + \beta_n \Delta_n, \lambda) - L(\theta_n, \lambda)) \mid \theta_n, \lambda \right] - \nabla^2_\theta L(\theta_n, \lambda) \right\| \to 0.$$

(ii) $\|E\left[\frac{1}{\beta_n}\Delta_n(L(\theta_n+\beta_n\Delta_n,\lambda)-L(\theta_n,\lambda))\mid\theta_n,\lambda\right]-\nabla L(\theta_n,\lambda)\|\to 0.$

Proof The proofs of the above claims follow from Propositions 10.1 and 10.2 of [17], respectively. ■

The rest of the analysis is identical to that of RS-SPSA-N. ■

Remark 11 (**On Convergence Rate.**) In the above, we established asymptotic limits for all our algorithms using the ODE approach. To the best of our knowledge, there are no convergence rate results available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms. This is true even for the actor-critic algorithms that do not incorporate any risk criterion. In [34], the authors provide asymptotic convergence rate results for *linear* two-timescale recursions. It would be an interesting direction for future research to obtain concentration bounds for general (non-linear) two-timescale schemes.

While a rigorous analysis on convergence rate of our proposed schemes is difficult, one could make a few concessions and use the following argument to see that the SPSA-based algorithms converge quickly: In order to analyse the rate of convergence of θ-recursion, assume (for sufficiently large n) that the TD-critic has converged in the inner-loop. This is because, the trajectory lengths $m_n\to\infty$ as $n\to\infty$ and under appropriate step-size settings (or with iterate averaging) one can obtain convergence rate of the order $O\left(1/\sqrt{m}\right)$ on the root mean square error of TD (see Theorem 1). Now, if one holds λ fixed, then invoking asymptotic normality results for SPSA (see Proposition 2 in [58]) it can be shown that $n^{1/3}(\theta_n-\theta^\lambda)$ is asymptotically normal, where θ^λ is a limit point in the set $\mathcal{Z}_\lambda$. Similar results also hold for second-order SPSA variants (cf. Theorem 3a in [60]). Both the aforementioned claims are proved using a well-known result on asymptotic normality of stochastic approximation schemes due to Fabian [27].

The second-order schemes such as RS-SPSA-N score over their first order counterpart RS-SPSA-G from a asymptotic normality results perspective. This is because obtaining the optimal convergence rate for RS-SPSA-G requires that the step-size $\zeta_2(n)$ is set to $\zeta_2(0)/n$ where $\zeta_2(0)>1/\lambda_{\min}(\nabla^2_\theta L(\theta^\lambda,\lambda))$, whereas there is no such constraint for the second-order algorithm RS-SPSA-N. Here $\lambda_{\min}(A)$ denotes the minimum eigenvalue of the matrix A. The reader is referred to [26] for a detailed discussion on convergence rate of (one timescale) SPSA-based schemes using asymptotic mean-square error.

Remark 12 (**Unstable Equilibria.**) The limit set $\mathcal{Z}_\lambda$ contains both stable and unstable equilibria and the θ-recursion can possibly end up in a unstable equilibrium point. One may avoid this situation by including additional noise in the randomized policy that drives the θ-recursion. For instance, define a η-offset policy as

$$\hat{\mu}(a\mid x)=\frac{\mu(a\mid x)+\eta}{\sum\limits_{a'\in\mathcal{A}(x)}(\mu(a'\mid x)+\eta)}.$$

The above policy can be used in place of the regular $\mu(\cdot\mid x)$, so that the algorithm is pulled away from an unstable equilibria. Providing theoretical guarantees for such a scheme is non-trivial and we have left it for future work.

8 Convergence Analysis of the Average Reward Risk-Sensitive Actor-Critic Algorithm

As in the discounted setting, we use the ODE approach [21] to analyze the convergence of our average reward risk-sensitive actor-critic algorithm. The proof involves three main steps:

1. The first step is the convergence of ρ, η, V, and U, for any fixed policy θ and Lagrange multiplier λ. This corresponds to a TD(0) (with extension to η and U) proof. Using arguments similar to that in Step 2 of the proof of RS-SPSA-G, one can show that the θ and λ recursions track $\dot{\theta}_t = 0$ and $\dot{\lambda}_t = 0$, when viewed from the TD critic timescale $\{\zeta_3(t)\}$. Thus, the policy θ and Lagrange multiplier λ are assumed to be constant in the analysis of the critic recursion.
2. The second step is to show the convergence of θ_n to an ε-neighborhood $\mathcal{Z}_\lambda^\varepsilon$ of the set of asymptotically stable equilibria $\mathcal{Z}_\lambda$ of ODE
$$\dot{\theta}_t = \check{\Gamma}\big(\nabla L(\theta_t, \lambda)\big), \tag{70}$$
where the projection operator $\check{\Gamma}$ ensures that the evolution of θ via the ODE (70) stays within the compact and convex set $\Theta \subset \mathbb{R}^{\kappa_1}$ and is defined in (56). Again here it is assumed that λ is fixed because θ-recursion is on a faster time-scale than λ's.
3. The final step is the convergence of λ and showing that the whole algorithm converges to a local saddle point of $L(\theta, \lambda)$. where the limit is shown to satisfy the variance constraint in (40).

Step 1: Critic's Convergence

Lemma 9 *For any given policy μ, $\{\widehat{\rho}_n\}$, $\{\widehat{\eta}_n\}$, $\{v_n\}$, and $\{u_n\}$, defined in Algorithm 2 and by the critic recursion* (46) *converge to $\rho(\mu)$, $\eta(\mu)$, v^μ, and u^μ almost surely, where v^μ and u^μ are the unique solutions to*

$$\Phi_v^\top \boldsymbol{D}^\mu \Phi_v v^\mu = \Phi_v^\top \boldsymbol{D}^\mu T_v^\mu(\Phi_v v^\mu), \qquad \Phi_u^\top \boldsymbol{D}^\mu \Phi_u u^\mu = \Phi_u^\top \boldsymbol{D}^\mu T_u^\mu(\Phi_u u^\mu), \tag{71}$$

respectively. In (71)*, $\boldsymbol{D}^\mu$ denotes the diagonal matrix with entries $d^\mu(x)$ for all $x \in \mathcal{X}$, and T_v^μ and T_u^μ are the Bellman operators for the differential value and square value functions of policy μ, defined as*

$$T_v^\mu J = \boldsymbol{r}^\mu - \rho(\mu)\boldsymbol{e} + \boldsymbol{P}^\mu J, \qquad T_u^\mu J = \boldsymbol{R}^\mu \boldsymbol{r}^\mu - \eta(\mu)\boldsymbol{e} + \boldsymbol{P}^\mu J, \tag{72}$$

where $\boldsymbol{r}^\mu$ and $\boldsymbol{P}^\mu$ are the reward vector and transition probability matrix of policy μ, $\boldsymbol{R}^\mu = diag(\boldsymbol{r}^\mu)$, and $\boldsymbol{e}$ is a vector of size n (the size of the state space $\mathcal{X}$) with elements all equal to one.

Proof The proof for the average reward $\rho(\mu)$ and differential value function v^μ follows in a similar manner as Lemma 5 in [14]. It is based on verifying the Assumptions (A1)-(A2) of Borkar and Meyn [23], and uses the second part of Assumption **(A3)** of our paper, i.e., $v \in \mathbb{R}^{\kappa_2}$, for every $v \in \mathbb{R}^{\kappa_2}$. The proof for $\rho(\mu)$ and v^μ can be easily extended to the square average reward $\eta(\mu)$ and square differential value function u^μ. ■

Step 2: Actor's Convergence

Let $\mathcal{Z}_\lambda = \big\{\theta \in \Theta : \check{\Gamma}\big(- \nabla L(\theta, \lambda)\big) = 0\big\}$ denote the set of asymptotically stable equilibrium points of the ODE (70) and $\mathcal{Z}_\lambda^\varepsilon = \big\{\theta \in \Theta : ||\theta - \theta_0|| < \varepsilon, \theta_0 \in \mathcal{Z}_\lambda\big\}$ denote the set of points in the ε-neighborhood of $\mathcal{Z}_\lambda$. The main result regarding the convergence of the policy parameter in (47) is as follows:

Theorem 7 *Assume (A1)-(A4). Then, given $\varepsilon > 0$, $\exists \beta > 0$ such that for θ_n, $n \geq 0$ obtained by the algorithm, if $\sup_{\theta_n} \|\mathcal{B}(\theta_n, \lambda)\| < \beta$, then θ_n governed by* (47) *converges almost surely to $\mathcal{Z}_\lambda^\varepsilon$ as $n \to \infty$.*

Proof Let $\mathcal{F}(n) = \sigma(\theta_m, m \leq n)$ denote a sequence of σ-fields. We have

$$\begin{aligned}
\theta_{n+1} &= \Gamma\Big(\theta_n - \zeta_2(n)\big(- \delta_n\psi_n + \lambda(\epsilon_n\psi_n - 2\widehat{\rho}_{n+1}\delta_n\psi_n)\big)\Big)\\
&= \Gamma\big(\theta_n + \zeta_2(n)(1 + 2\lambda\widehat{\rho}_{n+1})\delta_n\psi_n - \zeta_2(n)\lambda\epsilon_n\psi_n\big)\\
&= \Gamma\bigg(\theta_n - \zeta_2(n)\Big[1 + 2\lambda\Big(\big(\widehat{\rho}_{n+1} - \rho(\theta_n)\big) + \rho(\theta_n)\Big)\Big]\mathbb{E}\big[\delta^{\theta_n}\psi_n|\mathcal{F}(n)\big]\\
&\qquad - \zeta_2(n)\Big[1 + 2\lambda\Big(\big(\widehat{\rho}_{n+1} - \rho(\theta_n)\big) + \rho(\theta_n)\Big)\Big]\Big(\delta_n\psi_n - \mathbb{E}\big[\delta_n\psi_n|\mathcal{F}(n)\big]\Big)\\
&\qquad - \zeta_2(n)\Big[1 + 2\lambda\Big(\big(\widehat{\rho}_{n+1} - \rho(\theta_n)\big) + \rho(\theta_n)\Big)\Big]\mathbb{E}\big[(\delta_n - \delta^{\theta_n})\psi_n|\mathcal{F}(n)\big]\\
&\qquad + \zeta_2(n)\lambda\mathbb{E}\big[\epsilon^{\theta_n}\psi_n|\mathcal{F}(n)\big] + \zeta_2(n)\lambda\Big(\epsilon_n\psi_n - \mathbb{E}\big[\epsilon_n\psi_n|\mathcal{F}(n)\big]\Big)\\
&\qquad + \zeta_2(n)\lambda\mathbb{E}\big[(\epsilon_n - \epsilon^{\theta_n})\psi_n|\mathcal{F}(n)\big]\bigg).
\end{aligned}$$

By setting $\xi_n = \widehat{\rho}_{n+1} - \rho(\theta_n)$, we may write the above equation as

$$\begin{aligned}
\theta_{n+1} = \Gamma\bigg(&\theta_n - \zeta_2(n)\big[1 + 2\lambda\big(\xi_n + \rho(\theta_n)\big)\big]\mathbb{E}\big[\delta^{\theta_n}\psi_n|\mathcal{F}(n)\big] && (73)\\
&- \zeta_2(n)\big[1 + 2\lambda\big(\xi_n + \rho(\theta_n)\big)\big]\underbrace{\Big(\delta_n\psi_n - \mathbb{E}\big[\delta_n\psi_n|\mathcal{F}(n)\big]\Big)}_{*}\\
&- \zeta_2(n)\big[1 + 2\lambda\big(\xi_n + \rho(\theta_n)\big)\big]\underbrace{\mathbb{E}\big[(\delta_n - \delta^{\theta_n})\psi_n|\mathcal{F}(n)\big]}_{+}\\
&+ \zeta_2(n)\lambda\mathbb{E}\big[\epsilon^{\theta_n}\psi_n|\mathcal{F}(n)\big] + \zeta_2(n)\lambda\underbrace{\Big(\epsilon_n\psi_n - \mathbb{E}\big[\epsilon_n\psi_n|\mathcal{F}(n)\big]\Big)}_{*} && (74)\\
&+ \zeta_2(n)\lambda\underbrace{\mathbb{E}\big[(\epsilon_n - \epsilon^{\theta_n})\psi_n|\mathcal{F}(n)\big]}_{+}\bigg).
\end{aligned}$$

Since Algorithm 2 uses an unbiased estimator for ρ, we have $\widehat{\rho}_{n+1} \to \rho(\theta_n)$, and thus, $\xi_n \to 0$. The terms $(+)$ asymptotically vanish in light of Lemma 9 (Critic convergence). Finally the terms $(*)$ can be seen to vanish using standard martingale arguments (cf. Theorem 2 in [14]). Thus, (73) can be seen to be equivalent in an asymptotic sense to

$$\theta_{n+1} = \Gamma\Big(\theta_n - \zeta_2(n)\big[1 + 2\lambda\rho(\theta_n)\big]\mathbb{E}\big[\delta^{\theta_n}\psi_n|\mathcal{F}(n)\big] + \zeta_2(n)\lambda\mathbb{E}\big[\epsilon^{\theta_n}\psi_n|\mathcal{F}(n)\big]\Big). \quad (75)$$

From the foregoing, it can be seen that the actor recursion in (47) asymptotically tracks the stable fixed points of the ODE

$$\dot{\theta}_t = \check{\Gamma}\Big(\nabla L(\theta_t, \lambda) + \mathcal{B}(\theta_t, \lambda)\Big). \quad (76)$$

Note that the bias of Algorithm 2 in estimating $\nabla L(\theta, \lambda)$ is (see Lemma 5)

$$\begin{aligned}
\mathcal{B}(\theta, \lambda) = \sum_x \boldsymbol{D}^\theta(x)\Big\{ &- \big(1 + 2\lambda\rho(\theta)\big)\big[\nabla\bar{V}^\theta(x) - \nabla v^{\theta\top}\phi_v(x)\big]\\
&+ \lambda\big[\nabla\bar{U}^\theta(x) - \nabla u^{\theta\top}\phi_u(x)\big]\Big\}.
\end{aligned}$$

Since the bias $\sup_\theta \|\mathcal{B}(\theta,\lambda)\| \rightarrow 0$ by assumption, the trajectories (76) converge to those of (55) uniformly on compacts for the same initial condition and the claim follows. ■

Remark 13 (**Bias in Estimating Gradient.**) We do not always expect that $\sup_\theta \|\mathcal{B}(\theta,\lambda)\| \rightarrow 0$. However, if there is no bias or negligibly small bias in the actor-critic algorithm, which is directly related to the choice of the critic's function space, then we will definitely gain from using actor-critic instead of policy gradient. Note that the choice between actor-critic and policy gradient is a bias-variance tradeoff, and similar to any other bias-variance tradeoff, if the variance reduction is more significant (given the number of samples used to estimate each gradient) than the introduced bias, then it would be advantageous to use actor-critic instead of policy gradient. Also note that this tradeoff exists even in the original form (risk neutral) of actor-critic and policy gradient and has nothing to do with the risk-sensitive objective function studied in this paper. For more details on this, we refer the reader to Theorem 2 and Remark 2 in Bhatnagar et al [15].

Step 3: λ Convergence and Overall Convergence of the Algorithm

As in the discounted setting, we first show that the λ-recursion converges and then prove convergence to a local saddle point of $L(\theta,\lambda)$. Consider the ODE

$$\dot{\lambda}_t = \check{\Gamma}_\lambda\big(\Lambda(\theta^{\lambda_t}) - \alpha\big), \tag{77}$$

where $\check{\Gamma}_\lambda$ is a projection operator that forces the evolution of λ via (65) is within $[0,\lambda_{\max}]$ and is defined in (66).

Theorem 8 *$\lambda_n \rightarrow \mathcal{F}$ almost surely as $t \rightarrow \infty$, where $\mathcal{F} \stackrel{\triangle}{=} \big\{\lambda \mid \lambda \in [0,\lambda_{\max}], \check{\Gamma}_\lambda\big(\Lambda(\theta^\lambda) - \alpha\big) = 0,\ \theta^\lambda \in \mathcal{Z}_\lambda\big\}$.*

Proof The proof follows in a similar manner as that of Theorem 3 in [11]. ■

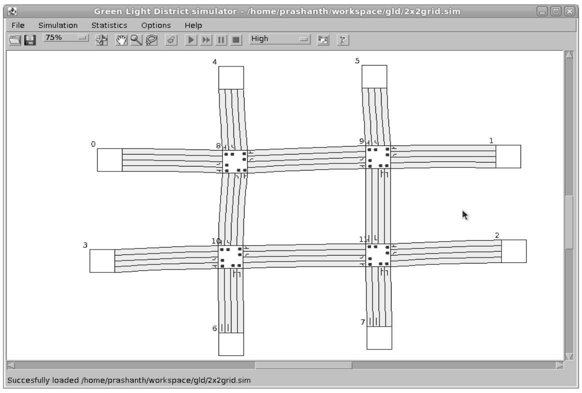
As in the discounted setting, the following proposition claims that the limit θ^{λ^*} corresponding to λ^* satisfies the variance constraint in (40), i.e.,

Proposition 2 *For any λ^* in $\hat{\mathcal{F}} \stackrel{\triangle}{=} \big\{\lambda \mid \lambda \in [0,\lambda_{\max}),\ \check{\Gamma}_\lambda\big[\Lambda^{\theta^\lambda}(x^0) - \alpha\big] = 0,\ \theta^\lambda \in \mathcal{Z}_\lambda\big\}$, the corresponding limiting point θ^{λ^*} satisfies the variance constraint $\Lambda^{\theta^{\lambda^*}}(x^0) \leq \alpha$.*

Using arguments similar to that used to prove convergence of RS-SPSA-G, it can be shown that that the ODE (77) is equivalent to $\dot{\lambda}_t = \check{\Gamma}_\lambda\big[\nabla_\lambda L(\theta^{\lambda_t},\lambda_t)\big]$ and thus, the actor parameters (θ_n,λ_n) updated according to (47) converge to a (local) saddle point $(\theta^{\lambda^*},\lambda^*)$ of $L(\theta,\lambda)$. Morever, the limiting point θ^{λ^*} satisfies the variance constraint in (40).

9 Experimental Results

We evaluate our algorithms in the context of a traffic signal control application. The objective in our formulation is to minimize the total number of vehicles in the system, which indirectly minimizes the delay experienced by the system. The motivation behind using a risk-sensitive control strategy is to reduce the variations in the delay experienced by road users.



Green Light District simulator - /home/prashanth/workspace/gld/2x2grid.sim

Succesfully loaded /home/prashanth/workspace/gld/2x2grid.sim

Fig. 2 The 2x2-grid network used in our traffic signal control experiments.

9.1 Implementation

We consider both infinite horizon discounted and average settings for the traffic signal control MDP, formulated as in [46]. We briefly recall their formulation here: The state at each time t, x_n, is the vector of queue lengths and elapsed times and is given by $x_n = (q_1(n), \ldots, q_N(n), t_1(n), \ldots, t_N(n))$, where N is the number of signalled lanes in the road network considered. Here q_i and t_i denote the queue length and elapsed time since the signal turned to red on lane i. The actions a_n belong to the set of feasible sign configurations. The single-stage cost function $h(x_n)$ is defined as follows:

$$\begin{aligned} h(x_n) = & r_1 * \big[\sum_{i \in I_p} r_2 * q_i(n) + \sum_{i \notin I_p} s_2 * q_i(n)\big] \\ & + s_1 * \big[\sum_{i \in I_p} r_2 * t_i(n) + \sum_{i \notin I_p} s_2 * t_i(n)\big], \end{aligned} \tag{78}$$

where $r_i, s_i \geq 0$ such that $r_i + s_i = 1$ for $i = 1, 2$ and $r_2 > s_2$. The set I_p is the set of prioritized lanes in the road network considered. While the weights r_1, s_1 are used to differentiate between the queue length and elapsed time factors, the weights r_2, s_2 help in prioritization of traffic.

Given the above traffic control setting, we aim to minimize both the long run discounted and average sum of the cost function $h(x_n)$ in (78). The underlying policy that guides the selection of the sign configuration in each of the algorithms we implemented (see below for the complete list) is a parameterized Boltzmann family and has the form

$$\mu_\theta(x, a) = \frac{e^{\theta^\top \phi_{x,a}}}{\sum_{a' \in \mathcal{A}(x)} e^{\theta^\top \phi_{x,a'}}}, \quad \forall x \in \mathcal{X}, \ \forall a \in \mathcal{A}. \tag{79}$$

The experiments for each algorithm that we implement is comprised of the following two phases:

Policy Search Phase: Here each iteration involved the simulation run with the nominal policy parameter θ as well as the perturbed policy parameter θ^+ (algorithm-specific). We run each algorithm for 500 iterations, where the run length for a particular policy parameter is 150 steps.

Policy Test Phase: After the completion of the policy search phase, we freeze the policy parameter and run 50 independent simulations with this (converged) choice of the parameter. The results presented subsequently are averages over these 50 runs.

We implement the following algorithms using the Green Light District (GLD) simulator [72][8]:

Discounted Setting

1. ***SPSA-G***: This is a first-order risk-neutral algorithm with SPSA-based gradient estimates that updates the parameter θ as follows:

$$\theta^{(i)}_{n+1} = \Gamma_i \left(\theta^{(i)}_n + \frac{\zeta_2(n)}{\beta \Delta^{(i)}_n} (v^+_n - v_n)^{\mathsf{T}} \phi_v(x^0) \right),$$

where the critic parameters v_n, v^+_n are updated according to (13). Note that this is a two-timescale algorithm with a TD critic on the faster timescale and the actor on the slower timescale. Unlike RS-SPSA-G, this algorithm, being risk-neutral, does not involve the Lagrange multiplier recursion.
2. ***SF-G***: This is a first-order risk-neutral algorithm that is similar to SPSA-G, except that the gradient estimation scheme used here is based on the smoothed functional (SF) technique. The update of the policy parameter in this algorithm is given by

$$\theta^{(i)}_{n+1} = \Gamma_i \left(\theta^{(i)}_n + \zeta_2(n) \Big(\frac{\Delta^{(i)}_n}{\beta} (v^+_n - v_n)^{\mathsf{T}} \phi_v(x^0) \Big) \right).$$

3. ***SPSA-N***: This is a risk-neutral algorithm and is the second-order counterpart of SPSA-G. The Hessian update in this algorithm is as follows: For $i, j = 1, \dots, \kappa_1$, $i < j$, the update is

$$H^{(i,j)}_{n+1} = H^{(i,j)}_n + \zeta'_2(n) \left[\frac{(v_n - v^+_n)^{\mathsf{T}} \phi_v(x^0)}{\beta^2 \Delta^{(i)}_n \widehat{\Delta}^{(j)}_n} - H^{(i,j)}_n \right], \tag{80}$$

and for $i > j$, we set $H^{(i,j)}_{n+1} = H^{(j,i)}_{n+1}$. As in RS-SPSA-N, let $M_n \triangleq H^{-1}_n$, where $H_n = \Upsilon\big([H^{(i,j)}_n]^{|\kappa_1|}_{i,j=1}\big)$. The actor updates the parameter θ as follows:

$$\theta^{(i)}_{n+1} = \Gamma_i \left[\theta^{(i)}_n + \zeta_2(n) \sum_{j=1}^{\kappa_1} M^{(i,j)}_n \Big(\frac{(v^+_n - v_n)^{\mathsf{T}} \phi_v(x^0)}{\beta \Delta^{(j)}_n} \Big) \right]. \tag{81}$$

The rest of the symbols, including the critic parameters, are as in RS-SPSA-N.
4. ***SF-N***: This is a risk-neutral algorithm and is the second-order counterpart of SF-G. It updates the Hessian and the actor as follows: For $i, j, k = 1, \dots, \kappa_1$, $j < k$, the Hessian update is

$$\textbf{Hessian:} \quad H^{(i,i)}_{n+1} = H^{(i,i)}_n + \zeta'_2(n) \left[\frac{\big(\Delta^{(i)^2}_n - 1\big)}{\beta^2} (v_n - v^+_n)^{\mathsf{T}} \phi_v(x^0) - H^{(i,i)}_n \right],$$

$$H^{(j,k)}_{n+1} = H^{(j,k)}_n + \zeta'_2(n) \left[\frac{\Delta^{(j)}_n \Delta^{(k)}_n}{\beta^2} (v_n - v^+_n)^{\mathsf{T}} \phi_v(x^0) - H^{(j,k)}_n \right],$$

[8] We would like to point out that the experimental setting involves ’costs’ and not ’rewards’ and the algorithms implemented should be understood as optimizing a negative reward.

and for $j > k$, we set $H_{n+1}^{(j,k)} = H_{n+1}^{(k,j)}$. As before, let $M_n \triangleq H_n^{-1}$, with H_n formed as in SPSA-N. Then, the actor update for the parameter θ is as follows:

$$\textbf{Actor:} \qquad \theta_{n+1}^{(i)} = \Gamma_i \left[\theta_n^{(i)} + \zeta_2(n) \sum_{j=1}^{\kappa_1} M_n^{(i,j)} \frac{\Delta_n^{(j)}}{\beta} (v_n^+ - v_n)^\mathsf{T} \phi_v(x^0) \right].$$

The rest of the symbols, including the critic parameters, are as in RS-SPSA-N.

5. ***RS-SPSA-G***: This is the first-order risk-sensitive actor-critic algorithm that attempts to solve (40) and updates according to (19).
6. ***RS-SF-G***: This is a first-order algorithm and the risk-sensitive variant of SF-G that updates the actor according to (20).
7. ***RS-SPSA-N***: This is a second-order risk-sensitive algorithm that estimates gradient and Hessian using SPSA and updates them according to (31).
8. ***RS-SF-N***: This second-order risk-sensitive algorithm is the SF counterpart of RS-SPSA-N, and updates according to (36).
9. ***TAMAR***: This is a straightforward adaptation of the algorithm proposed in [68]. The main difference between this and our algorithms is that `TAMAR` uses a Monte Carlo critic, while our algorithms employ a TD critic. Moreover, `TAMAR` incorporates the λ-recursion that is identical to that of our algorithms (see Eq. 21). In contrast, the algorithm proposed in [68] is for a fixed λ that may not be optimal. Note that even though `TAMAR` is an algorithm proposed for a stochastic shortest path (SSP) setting, it can be implemented in the traffic signal control problem since we truncate the simulation after 150 steps.
   Let D_n denote the sum of rewards obtained from a single simulation run in the policy search phase. Further, let $z_n := \sum_{m=0}^{150} \nabla \ln \mu_\theta(x_m, a_m)$ denote the likelihood derivative. Then, the update rule is given by

$$\begin{aligned}
\tilde{V}_{n+1} =& \tilde{V}_n + \zeta_3(n)\big(D_n - \tilde{V}_n\big)\\
\tilde{\Lambda}_{n+1} =& \tilde{\Lambda}_n + \zeta_3(n)\big(D_n^2 - \tilde{V}_n^2 - \tilde{\Lambda}_n\big)\\
\theta_{n+1}^{(i)} =& \Gamma_i \left(\theta_n + \zeta_2(n)\big(D_n - \lambda_n(D_n^2 - 2D_n\tilde{V}_n)\big) z_n^{(i)} \right), i = 1, \ldots, \kappa_1,\\
\lambda_{n+1} =& \Gamma_\lambda \left[\lambda_n + \zeta_1(n)\Big(\Lambda_n - \alpha\Big) \right].
\end{aligned}$$

   Note that the θ-recursion above corrects an error (we believe it is a typo) in the corresponding update rule (i.e., Eq. 13 in [68]). Unlike the above, Eq. 13 in [68] is missing the multiplier D_n in the last term in the θ-recursion. The latter multiplier originates from the gradient of the value function (see Lemma 4.2 in [68]).

Average Setting

1. ***AC***: This is an actor-critic algorithm that minimizes the long-run average sum of the single-stage cost function $h(x_n)$, without considering any risk criteria. This is similar to Algorithm 1 in Bhatnagar et al [14].
2. ***RS-AC***: This is the risk-sensitive actor-critic algorithm that attempts to solve (40) and is described in Section 6.

All our algorithms incorporate function approximation owing to the curse of dimensionality associated with larger road networks. For instance, assuming only 20 vehicles per lane of a 2x2-grid network, the cardinality of the state space is approximately of the order 10^{32} and the situation is aggravated as the size of the road network increases. We employ the feature selection scheme from [47] in each of our algorithms. The features are obtained with coarse congestion estimates along the lanes of the road network as input. For instance, instead of the exact queue length on a lane, the coarse congestion information specifies whether the queue length was between 0 to L_1 units, between L_1 and L_2 units or greater than L_2 units. By placing magnetic sensor loops on the lane at distances L_1 and L_2 from the junction, it is possible to obtain coarse congestion information. Assume another threshold T_1 for the elapsed time. Using the aforementioned coarse inputs on queue lengths and elapsed times for each lane in the road network considered, the feature selection is performed in a graded fashion as follows: queue length less than L_1 and elapsed time less than T_1 leading a to feature value that recommends red light, queue length more than L_2 and elapsed time more than T_1 leading to a feature value that recommends green light, with the feature values for the intermediate scenarios graded appropriately. For a detailed description of the feature selection scheme, the reader is referred to Section V-B of [47]. The values L_1, L_2 and T_1 are set to 6, 14 and 130, as recommended in [47].

Figure 2 shows a snapshot of the road network used for conducting the experiments from GLD simulator. Traffic is added to the network at each time step from the edge nodes. The spawn frequencies specify the rate at which traffic is generated at each edge node and follow a Poisson distribution. The spawn frequencies are set such that the proportion of the number of vehicles on the main roads (the horizontal ones in Fig. 2) to those on the side roads is in the ratio of $100 : 5$. This setting is close to what is observed in practice and has also been used for instance in [46, 47]. In all our experiments, we set the weights in the single stage cost function (78) as follows: $r_1 = r_2 = 0.5$ and $r_2 = 0.6, s_2 = 0.4$. For the SPSA and SF-based algorithms in the discounted setting, we set the parameter $\delta = 0.2$ and the discount factor $\gamma = 0.9$. The parameter α in the formulations (40) and (3) was set to 20. The step-size sequences are chosen as follows:

$$\zeta_1(n) = \frac{1}{n}, \quad \zeta_2(n) = \frac{1}{n^{0.75}}, \quad \zeta_2'(n) = \frac{1}{n^{0.7}}, \quad \zeta_3(n) = \frac{1}{n^{0.66}}, \qquad n \geq 1. \tag{82}$$

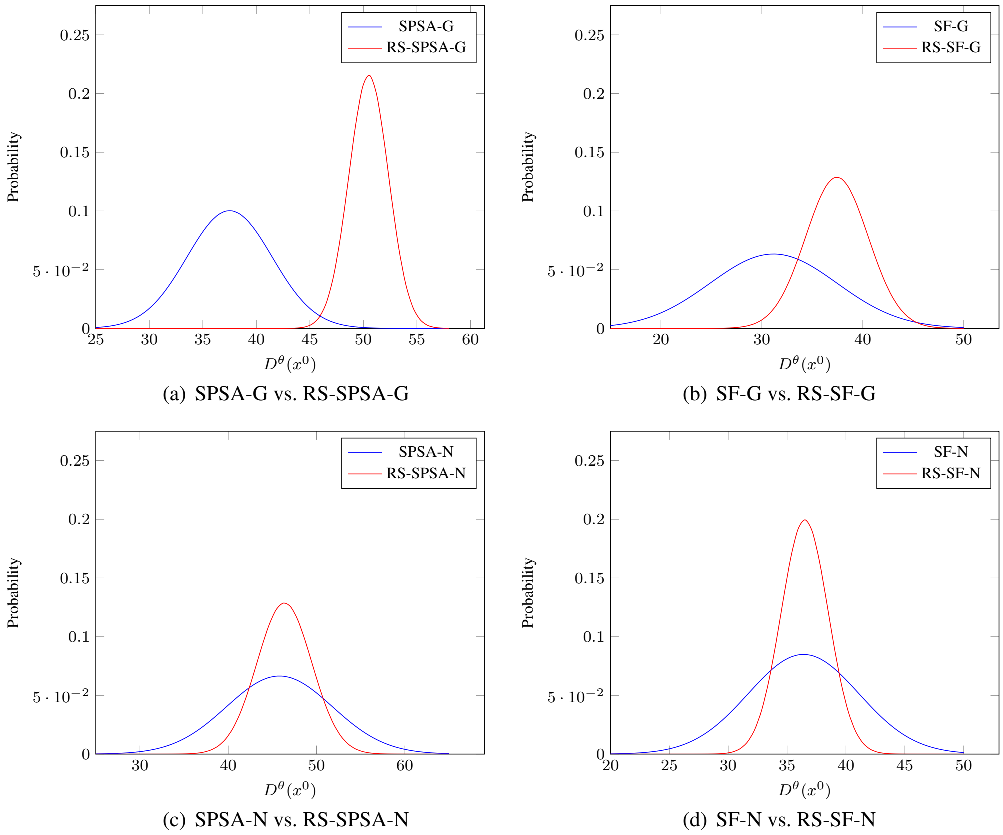
Further, the constant k related to $\zeta_4(n)$ in the risk-sensitive average reward algorithm is set to 1. It is easy to see that the choice of step-sizes above satisfies (A4). The projection operator $\varGamma_i$ was set to project the iterate $\theta^{(i)}$ onto the set $[0, 10]$, for all $i = 1, \ldots, \kappa_1$, while the projection operator for the Lagrange multiplier used the set $[0, 1000]$. The initial policy parameter θ_0 was set to the κ_1-dimensional vector of ones. All the experiments were performed on a 2.53GHz Intel quad core machine with 3.8GB RAM.

9.2 Results

Figure 3 shows the distribution of the discounted cumulative cost $D^\theta(x^0)$ for the algorithms in the discounted setting. Figure 4 shows the total arrived road users (TAR) obtained for all the algorithms in the discounted setting, whereas Figure 5 presents the average junction waiting time (AJWT) for the first-order SF-based algorithm RS-SF-G.[9] TAR is a throughput metric that measures the number of road users who have reached their destination, whereas AJWT is a delay metric that quantifies the average delay experienced by the road users.

The performance of the algorithms in the average setting is presented in Figure 6. In particular, Figure 6(a) shows the distribution of the average reward ρ, while Figure 6(b) presents the average junction waiting time (AJWT) for the average cost algorithms.

[9] The AJWT performance of the other algorithms in the discounted setting is similar and the corresponding plots are omitted here.



(a) SPSA-G vs. RS-SPSA-G

(b) SF-G vs. RS-SF-G

(c) SPSA-N vs. RS-SPSA-N

(d) SF-N vs. RS-SF-N

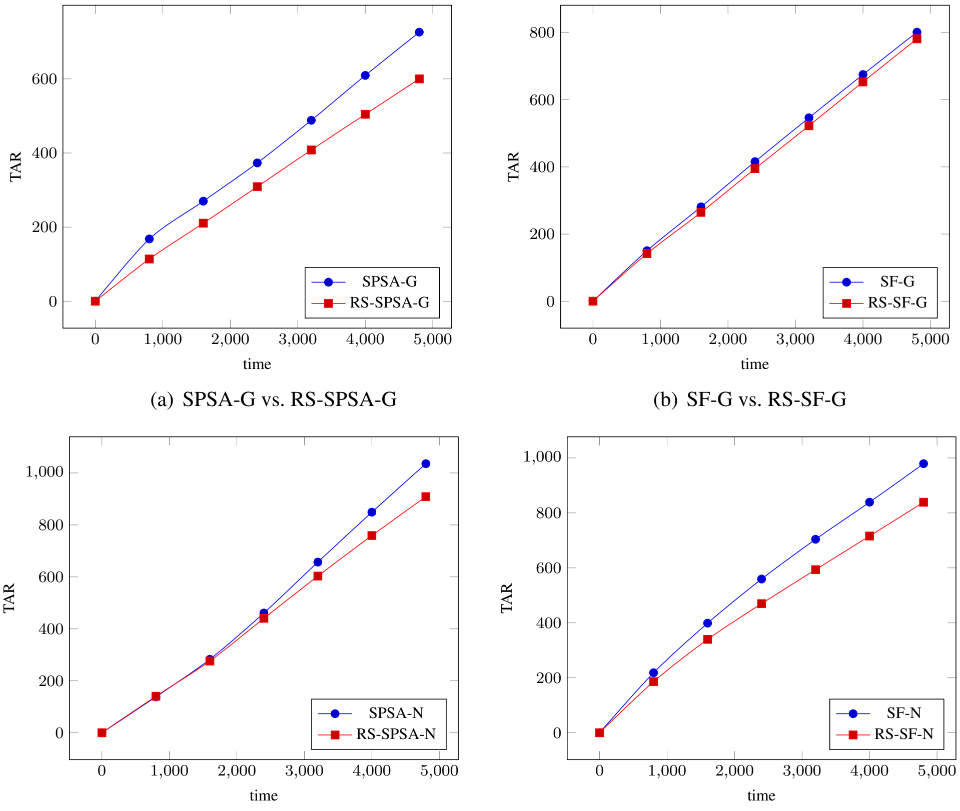
Fig. 3 Performance comparison in the discounted setting using the distribution of $D^{\theta}(x^0)$.

Table 1 Throughput (TAR) for algorithms in the discounted setting: standard deviation from 50 independent simulations shown after ±

Algorithm	Risk-neutral	Risk-sensitive
SPSA-G	754.84 ± 317.06	622.38 ± 28.36
SF-G	832.34 ± 82.24	810.82 ± 36.56
SPSA-N	$1077.2.66 \pm 250.42$	942.3 ± 65.77
SF-N	1013.62 ± 152.22	870.5 ± 61.61

Observation 1: *Risk-sensitive algorithms that we propose result in a long-term (discounted or average) cost that is higher than their risk-neutral variants, but with a significantly lower empirical variance of the cost in both discounted as well as average cost settings.*

The above observation is apparent from Figures 3 and 6(a), which present results for discounted and average cost settings respectively.



(a) SPSA-G vs. RS-SPSA-G

(b) SF-G vs. RS-SF-G

(c) SPSA-N vs. RS-SPSA-N

(d) SF-N vs. RS-SF-N

Fig. 4 Performance comparison of the algorithms in the discounted setting using the total arrived road users (TAR).

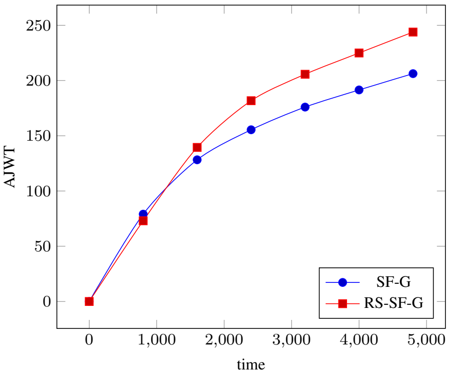
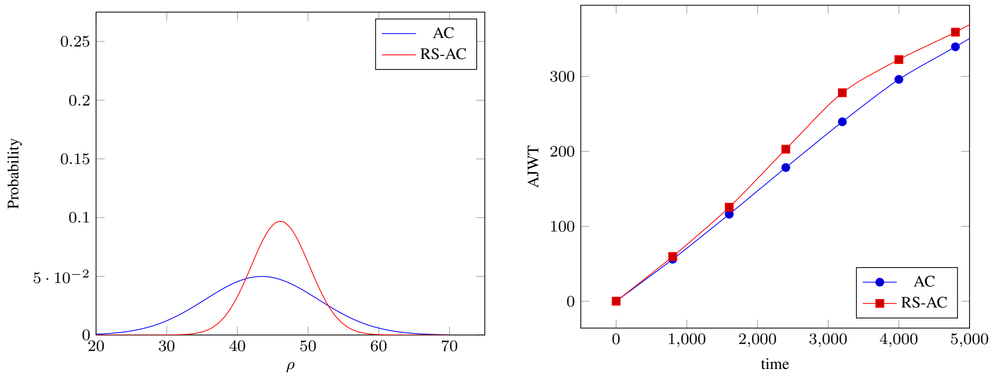


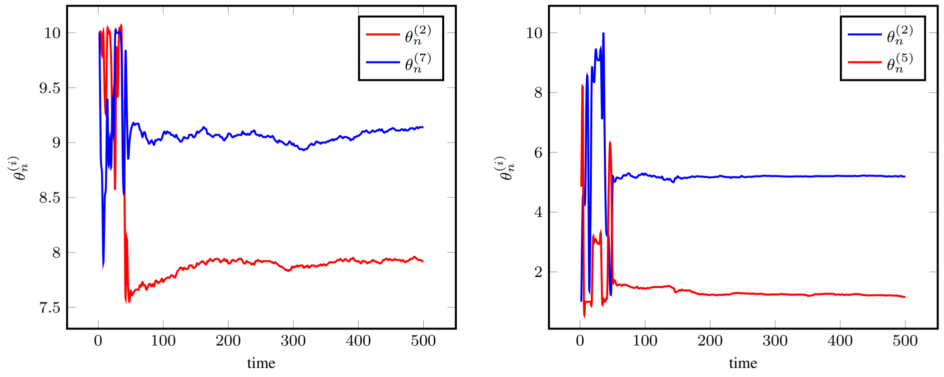
Fig. 5 Performance comparison of the first-order SF-based algorithms, SF-G and RS-SF-G, using the average junction waiting time (AJWT).



(a) average reward ρ distribution

(b) average junction waiting time

Fig. 6 Performance comparison of the risk-neutral (AC) and risk-sensitive (RS-AC) average reward actor-critic algorithms using two different metrics.



(a) RS-SPSA-G

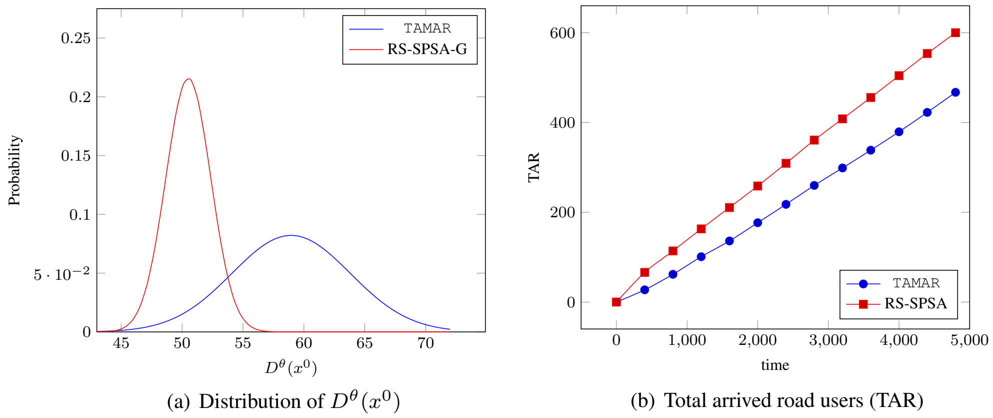
(b) RS-SPSA-N

Fig. 7 Convergence of SPSA based algorithms in the discounted setting – illustration using two (arbitrarily chosen) coordinates of the parameter θ.

Observation 2: *From a traffic signal control application standpoint, the risk-sensitive algorithms exhibit a mean throughput/delay that is close to that of the corresponding risk-neutral algorithms, but with a lower empirical variance in throughput/delay.*

Figures 4, 5, and 6(b) validate the first part of the observation above, while the results for the discounted risk-sensitive algorithms in Table 1 substantiate the second part in the above observation. In particular, Table 1 presents the mean and standard deviation of the final TAR value (i.e., the TAR value observed at the end of the policy test phase) for both first-order and second-order algorithms in the discounted setting and it is evident that the risk-sensitive algorithms exhibit a lower empirical variance in TAR when compared to their risk-neutral counterparts.

From the results in Figures 3–4 and Table 1, it is apparent that the second-order schemes (RS-SPSA-N and RS-SF-N) in the discounted setting exhibit better results in comparison to



(a) Distribution of $D^\theta(x^0)$ (b) Total arrived road users (TAR)

Fig. 8 Performance comparison of RS-SPSA and TAMAR [68] algorithms using two different metrics.

Table 2 ℓ_2 distance between gradient estimated using either RS-SPSA or TAMAR and a likelihood ratio benchmark: mean and standard error from 100 replications shown before and after ±, respectively

Policy	TAMAR	**RS-SPSA**
$\theta^{(i)} = 0.5, \forall i$	655.77 ± 18.65	142.1 ± 9.56
$\theta^{(i)} = 1, \forall i$	694.99 ± 16.67	149.82 ± 10.25
$\theta^{(i)} = 2, \forall i$	720.99 ± 14.85	146.67 ± 9.31
$\theta^{(i)} = 5, \forall i$	941.53 ± 25.39	200.08 ± 13.25
$\theta^{(i)} = 7, \forall i$	1167.78 ± 37.14	210.73 ± 12.97
$\theta^{(i)} = 10, \forall i$	1489.32 ± 43.43	277.15 ± 11.93

first-order methods (RS-SPSA-G and RS-SF-G), from the mean and variance of the long-term discounted cost as well as the throughput (TAR) performance.

Observation 3: *The policy parameter θ converges for the risk-sensitive algorithms.*

The above observation is validated for SPSA based algorithms in the discounted setting in Figures 7(a) and 7(b). Note that we established theoretical convergence of our algorithms earlier (see Sections 7 and 8) and these plots confirm the same. Further, these plots also show that the transient period, i.e., the initial phase when θ has not converged, is short. Similar observations hold for the other algorithms as well. The results of this section indicate the rapid empirical convergence of our proposed algorithms. This observation coupled with the fact that they guarantee low variance of return, make them attractive for implementation in risk-constrained systems.

Observation 4: *RS-SPSA, which is based on an actor-critic architecture, outperforms TAMAR, which employs a policy gradient approach.*

Figure 8 shows the distribution of the cumulative cost $D^\theta(x^0)$ and the total arrived road users (TAR) obtained for TAMAR and RS-SPSA algorithms. It is evident that RS-SPSA performs better than TAMAR in terms of mean as well as variance of the cumulative cost and also in terms of the throughput (TAR) observed. These results illustrate the benefits of using an actor-critic architecture. Note that both algorithms use the same parameterized Boltzmann policy (see Eq. 79) and the results have been obtained with the same number of updates, i.e., 500 SPSA updates, which is equivalent to 1000 policy gradient updates, as each iteration of SPSA uses two trajectories to estimate the gradient. While the results in Figure 8 implicitly indicate that RS-SPSA gives a better estimate of the gradient in comparison to TAMAR, we make this observation explicit in Table 2, which plots the results from the following experiment:

Step 1 (True gradient estimation): Estimate $\nabla_\theta \Lambda(x^0)$ using the likelihood ratio method, along the lines of Lemma 4.2 in [68]. For this purpose, simulate a large number, say $\top_1 = 1000$, of trajectories of the underlying MDP (as before, we truncate the trajectories to 150 steps). This estimate can be safely assumed to be very close to the true gradient and hence, we shall use it as the benchmark for comparing our SPSA based actor-critic scheme vs. the policy gradient approach of TAMAR.

Step 2 (Policy gradient approach of TAMAR):

- Fix a policy parameter.
- Run two simulations for the policy above.
- Estimate $\nabla_\theta \Lambda(x^0)$ using the scheme in TAMAR.
- Calculate the distance (in ℓ_2 norm) between the estimate above and the benchmark defined in Step 1.

Repeat the above steps 100 times and collect the mean and standard errors of the ℓ_2 distance in the last step above.

Step 3 (Actor-critic approach of RS-SPSA):

- Fix a policy parameter.
- Run two simulations - one for the unperturbed parameter and the another for the perturbed parameter, where perturbation is performed as in RS-SPSA (see Section 4.3).
- Estimate $\nabla_\theta \Lambda(x^0)$ using the scheme in RS-SPSA.
- Calculate the distance (in ℓ_2 norm) between the estimate above and the benchmark defined in Step 1.

Repeat the above steps 100 times and collect the mean and standard errors of the relevant ℓ_2 distance as in Step 2.

From the mean and standard errors presented in Table 2 for six different policies, it is evident that RS-SPSA produces more accurate estimates of the policy gradients than TAMAR, which explains its faster convergence (compared to TAMAR) in the experiments of Figure 8. The trend did not change by having the true gradient estimated from a larger number of trajectories. In particular, with $\top_1 = 5000$ (see Step 1 above), the relevant ℓ_2 distances for TAMAR and RS-SPSA were observed to be (683.06 ± 26.75) and (143.02 ± 14.44), respectively for the policy $\theta^{(i)} = 1, \forall i$.

10 Conclusions and Future Work

We proposed novel actor-critic algorithms for control in risk-sensitive discounted and average reward MDPs. All our algorithms involve a TD critic on the fast timescale, a policy gradient (actor) on the intermediate timescale, and a dual ascent for Lagrange multipliers on the slowest timescale. In the discounted setting, we pointed out the difficulty in estimating the gradient of the variance of the return and incorporated simultaneous perturbation based SPSA and SF approaches for gradient estimation in our algorithms. The average setting, on the other hand, allowed for an actor to employ compatible features to estimate the gradient of the variance. We provided proofs of convergence to locally (risk-sensitive) optimal policies for all the proposed algorithms. Further, using a traffic signal control application, we observed that our algorithms resulted in lower variance empirically as compared to their risk-neutral counterparts.

As future work, it would be interesting to develop a risk-sensitive algorithm that uses a single trajectory in the discounted setting. An orthogonal direction of future research is to obtain finite-time bounds on the quality of the solution obtained by our algorithms. As mentioned earlier, this is challenging as, to the best of our knowledge, there are no convergence rate results available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms.

References

1. Altman E (1999) Constrained Markov decision processes, vol 7. CRC Press
2. Barto A, Sutton R, Anderson C (1983) Neuron-like elements that can solve difficult learning control problems. IEEE Transaction on Systems, Man and Cybernetics 13:835–846
3. Basu A, Bhattacharyya T, Borkar V (2008) A learning algorithm for risk-sensitive cost. Mathematics of Operations Research 33(4):880–898
4. Baxter J, Bartlett P (2001) Infinite-horizon policy-gradient estimation. Journal of Artificial Intelligence Research 15:319–350
5. Bertsekas D (1995) Dynamic Programming and Optimal Control. Athena Scientific
6. Bertsekas D (1999) Nonlinear programming. Athena Scientific
7. Bertsekas D, Tsitsiklis J (1996) Neuro-Dynamic Programming. Athena Scientific
8. Bhatnagar S (2005) Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. ACM Transactions on Modeling and Computer Simulation 15(1):74–107
9. Bhatnagar S (2007) Adaptive Newton-based multivariate smoothed functional algorithms for simulation optimization. ACM Transactions on Modeling and Computer Simulation 18(1):1–35
10. Bhatnagar S (2010) An actor–critic algorithm with function approximation for discounted cost constrained Markov decision processes. Systems & Control Letters 59(12):760–766
11. Bhatnagar S, Lakshmanan K (2012) An online actor-critic algorithm with function approximation for constrained Markov decision processes. Journal of Optimization Theory and Applications pp 1–21
12. Bhatnagar S, Fu M, Marcus S, Wang I (2003) Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. ACM Transactions on Modeling and Computer Simulation 13(2):180–209
13. Bhatnagar S, Sutton R, Ghavamzadeh M, Lee M (2007) Incremental natural actor-Critic algorithms. In: Proceedings of Advances in Neural Information Processing Systems 20, pp 105–112
14. Bhatnagar S, Sutton R, Ghavamzadeh M, Lee M (2009) Natural actor-critic algorithms. Automatica 45(11):2471–2482
15. Bhatnagar S, Sutton R, Ghavamzadeh M, Lee M (2009) Natural actor-critic algorithms. Tech. Rep. TR09-10, Department of Computing Science, University of Alberta

16. Bhatnagar S, Hemachandra N, Mishra V (2011) Stochastic approximation algorithms for constrained optimization via simulation. ACM Transactions on Modeling and Computer Simulation 21(3):15
17. Bhatnagar S, Prasad H, Prashanth L (2013) Stochastic Recursive Algorithms for Optimization, vol 434. Springer
18. Borkar V (2001) A sensitivity formula for the risk-sensitive cost and the actor-critic algorithm. Systems & Control Letters 44:339–346
19. Borkar V (2002) Q-learning for risk-sensitive control. Mathematics of Operations Research 27:294–311
20. Borkar V (2005) An actor-critic algorithm for constrained Markov decision processes. Systems & Control Letters 54(3):207–213
21. Borkar V (2008) Stochastic approximation: a dynamical systems viewpoint. Cambridge University Press
22. Borkar V (2010) Learning algorithms for risk-sensitive control. In: Proceedings of the Nineteenth International Symposium on Mathematical Theory of Networks and Systems, pp 1327–1332
23. Borkar VS, Meyn SP (2000) The ode method for convergence of stochastic approximation and reinforcement learning. SIAM Journal on Control and Optimization 38(2):447–469
24. Chen H, Duncan T, Pasik-Duncan B (1999) A Kiefer-Wolfowitz algorithm with randomized differences. IEEE Transactions on Automatic Control 44(3):442–453
25. Delage E, Mannor S (2010) Percentile optimization for Markov decision processes with parameter uncertainty. Operations Research 58(1):203–213
26. Dippon J, Renz J (1997) Weighted means in stochastic approximation of minima. SIAM Journal on Control and Optimization 35(5):1811–1827
27. Fabian V (1968) On asymptotic normality in stochastic approximation. The Annals of Mathematical Statistics pp 1327–1332
28. Filar J, Kallenberg L, Lee H (1989) Variance-penalized Markov decision processes. Mathematics of Operations Research 14(1):147–161
29. Filar J, Krass D, Ross K (1995) Percentile performance criteria for limiting average Markov decision processes. IEEE Transaction of Automatic Control 40(1):2–10
30. Gill P, Murray W, Wright M (1981) Practical optimization. Academic press
31. Howard R, Matheson J (1972) Risk sensitive Markov decision processes. Management Science 18(7):356–369
32. Katkovnik V, Kulchitsky Y (1972) Convergence of a class of random search algorithms. Automatic Remote Control 8:81–87
33. Konda V, Tsitsiklis J (2000) Actor-Critic algorithms. In: Proceedings of Advances in Neural Information Processing Systems 12, pp 1008–1014
34. Konda VR, Tsitsiklis JN (2004) Convergence rate of linear two-time-scale stochastic approximation. Annals of Applied Probability pp 796–819
35. Korda N, Prashanth L (2015) On TD (0) with function approximation: Concentration bounds and a centered variant with exponential convergence. In: International Conference on Machine Learning (ICML)
36. Kushner H, Clark D (1978) Stochastic approximation methods for constrained and unconstrained systems. Springer-Verlag
37. Mannor S, Tsitsiklis J (2011) Mean-variance optimization in Markov decision processes. In: Proceedings of the Twenty-Eighth International Conference on Machine Learning, pp 177–184

38. Mannor S, Tsitsiklis JN (2013) Algorithmic aspects of mean–variance optimization in markov decision processes. European Journal of Operational Research 231(3):645–653
39. Marbach P (1998) Simulated-based methods for Markov decision processes. PhD thesis, Massachusetts Institute of Technology
40. Mas-Colell A, Whinston M, Green J (1995) Microeconomic theory. Oxford University Press
41. Mihatsch O, Neuneier R (2002) Risk-sensitive reinforcement learning. Machine Learning 49(2):267–290
42. Milgrom P, Segal I (2002) Envelope theorems for arbitrary choice sets. Econometrica 70(2):583–601
43. Nilim A, Ghaoui LE (2005) Robust control of Markov decision processes with uncertain transition matrices. Operations Research 53(5):780–798
44. Peters J, Vijayakumar S, Schaal S (2005) Natural actor-critic. In: Proceedings of the Sixteenth European Conference on Machine Learning, pp 280–291
45. Polyak BT, Juditsky AB (1992) Acceleration of stochastic approximation by averaging. SIAM Journal on Control and Optimization 30(4):838–855
46. Prashanth L, Bhatnagar S (2011) Reinforcement Learning With Function Approximation for Traffic Signal Control. IEEE Transactions on Intelligent Transportation Systems 12(2):412 –421
47. Prashanth L, Bhatnagar S (2012) Threshold Tuning Using Stochastic Optimization for Graded Signal Control. IEEE Transactions on Vehicular Technology 61(9):3865 –3880
48. Prashanth L, Ghavamzadeh M (2013) Actor-critic algorithms for risk-sensitive MDPs. In: Proceedings of Advances in Neural Information Processing Systems 26, pp 252–260
49. Prashanth L, Cheng J, Fu M, Marcus S (2015) Cumulative Prospect Theory Meets Reinforcement Learning: Prediction and Control. arXiv preprint arXiv:150602632v2
50. Puterman M (1994) Markov decision processes: Discrete stochastic dynamic programming. John Wiley & Sons
51. Ruppert D (1991) Stochastic approximation. Handbook of Sequential Analysis pp 503–529
52. Ruszczyński A (2010) Risk-averse dynamic programming for Markov decision processes. Mathematical Programming 125:235–261
53. Schweitzer PJ (1968) Perturbation theory and finite Markov chains. Journal of Applied Probability pp 401–413
54. Sharpe W (1966) Mutual fund performance. Journal of Business 39(1):119–138
55. Shen Y, Stannat W, Obermayer K (2013) Risk-sensitive Markov control processes. SIAM Journal on Control and Optimization 51(5):3652–3672
56. Sion M, et al (1958) On general minimax theorems. Pacific J Math 8(1):171–176
57. Sobel M (1982) The variance of discounted Markov decision processes. Applied Probability pp 794–802
58. Spall J (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control 37(3):332–341
59. Spall J (1997) A one-measurement form of simultaneous perturbation stochastic approximation. Automatica 33(1):109–112
60. Spall J (2000) Adaptive stochastic approximation by the simultaneous perturbation method. IEEE Transactions on Automatic Control 45(10):1839–1853
61. Styblinski MA, Opalski LJ (1986) Algorithms and software tools for IC yield optimization based on fundamental fabrication parameters. IEEE Transactions on Computer Aided Design CAD 1(5):79–89

62. Sutton R (1984) Temporal credit assignment in reinforcement learning. PhD thesis, University of Massachusetts Amherst
63. Sutton R (1988) Learning to predict by the methods of temporal differences. Machine Learning 3:9–44
64. Sutton R, Barto A (1998) Reinforcement learning: An introduction. MIT Press
65. Sutton R, McAllester D, Singh S, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. In: Proceedings of Advances in Neural Information Processing Systems 12, pp 1057–1063
66. Sutton RS, McAllester DA, Singh SP, Mansour Y, et al (1999) Policy gradient methods for reinforcement learning with function approximation. In: NIPS, Citeseer, vol 99, pp 1057–1063
67. Tamar A, Mannor S (2013) Variance adjusted actor-critic algorithms. arXiv preprint arXiv:13103697
68. Tamar A, Di Castro D, Mannor S (2012) Policy gradients with variance related risk criteria. In: Proceedings of the Twenty-Ninth International Conference on Machine Learning, pp 387–396
69. Tamar A, Di Castro D, Mannor S (2013) Policy evaluation with variance related risk criteria in markov decision processes. arXiv preprint arXiv:13010104
70. Tamar A, Di Castro D, Mannor S (2013) Temporal difference methods for the variance of the reward to go. In: Proceedings of the Thirtieth International Conference on Machine Learning, pp 495–503
71. Tsitsiklis JN, Van Roy B (1997) An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control 42(5):674–690
72. Wiering M, Vreeken J, van Veenen J, Koopman A (2004) Simulation and optimization of traffic in a city. In: IEEE Intelligent Vehicles Symposium, pp 453–458
73. Williams R (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning 8:229–256
74. Xu H, Mannor S (2012) Distributionally robust Markov decision processes. Mathematics of Operations Research 37(2):288–300