# Exponential Bellman Equation: A Distributional View of Risk-Sensitive Reinforcement Learning

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Risk sensitivity plays a critical role in practical reinforcement learning (RL) problems. Although risk sensitivity is often a byproduct of distributional RL, most existing algorithms compute risk-sensitive policies in a heuristic way and generally lack theoretical guarantees. In this paper, we study exponential Bellman equations, which relate the current and future cumulative rewards in a multiplicative manner. In particular, they yield an alternative representation of the cumulative reward distribution via the moment generating function. They further allow principled learning of optimal policies for RL with the entropic risk measure, a foundational framework of risk-sensitive RL. Our approach therefore provides a more direct route connecting distributional and risk-sensitive RL. In addition, we leverage the exponential Bellman equations to design and analyze two risk-sensitive RL algorithms with a novel exploration mechanism. Compared to existing algorithms, our algorithms achieve regret bounds that enjoy exponential improvements in the episode length and risk sensitivity.

## 1 Introduction

Risk-sensitive reinforcement learning (RL) is important for practical and high-stake applications, such as medicine and self-driving. It optimizes some risk measure of cumulative rewards, instead of their expectation as in risk-neutral RL. One popular approach to implementing risk-sensitive RL is through distributional RL [3, 26], which aims to learn policies based on the distribution of rewards and thus goes beyond expected rewards. Typically, distributional RL algorithms alternate between estimating the reward distribution via some functionals of the distribution, e.g. quantiles or density, and constructing policies from the distribution estimates. Unfortunately, despite their promising empirical results, distributional RL algorithms generally lack theoretical guarantees for the derived risk-sensitive policies. A main diffculty lies in deriving principled policy improvement procedures for risk-sensitive learning directly from Bellman equations. As a result, most distributional RL algorithms resort to heuristics for learning risk-sensitive policies. This poses great challenges to analyzing theoretical properties, especially non-asymptotic ones such as sample efficiency, of the derived risk-sensitive policies.

We provide a new perspective on risk-sensitive RL and its connection with distributional RL by considering exponential Bellman equations, which can be informally written as

$$\forall \mu \in \mathbb{R}, \quad e^{\mu \cdot Q} = \mathbb{E}_P[e^{\mu(r+V')}]. \tag{1}$$

Here, $r$ denotes a reward, $P$ is the joint distribution of state transition and the reward, and $Q$ and $V'$ are the current and future values (or cumulative utilities), respectively. We remark that (1) is distinct from the result of transforming the typical Bellman equation $\tilde{Q} = \mathbb{E}_P[r + \tilde{V}']$ by multiplying both sides with $\mu$ and raising them to a power of $e$, given that its expectation is taken on $e^{\mu(r+V')}$. It can

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

be seen that the reward and future value in Equation (1) are related in a multiplicative way, rather than the additive manner as in standard Bellman equations. We also see that $e^{\mu \cdot Q}$ represents the moment generating function (MGF) of the cumulative rewards. For distributional RL, this implies that one may obtain an alternative representation of the value distribution by estimating the MGF, which uniquely identifies a distribution just like distribution functions and quantiles.

One advantage of the exponential Bellman equations is that they naturally yield principled policy improvement for risk-sensitive RL by taking the greedy policy with respect to the quantity $e^{\beta \cdot Q}$. Specifically, the policy improvement procedure leads to risk-sensitive policies with respect to the entropic risk measure [19]. A foundational framework for risk-sensitive decision-making, RL with the entropic risk measure maximizes the objective

$$V^{\pi} = \beta^{-1} \log\{\mathbb{E}_P[e^{\beta \cdot R}]\},$$

with respect to the policy $\pi$, where $\beta \neq 0$ is a given risk parameter and $R$ denotes the cumulative rewards. It can be seen that the quantity $e^{\beta \cdot V^{\pi}}$ is the MGF (evaluated at $\beta$) of the reward distribution, which plays a similar role to $e^{\mu \cdot Q}$ of Equation (1). Therefore, Equation (1) serves as a direct bridge between distributional and risk-sensitive RL.

From the exponential Bellman equations, we develop a novel analysis of the Bellman backup procedure for risk-sensitive RL algorithms that are based on optimism. The analysis further motivates a novel exploration design called *doubly decaying* bonus, which helps the algorithms adapt to their estimation error over each horizon step while at the same time exploring efficiently. These discoveries enable us to propose two model-free algorithms for RL with the entropic risk measure based on the novel bonus. By combining the new analysis and bonus design, we prove that the preceding algorithms attain nearly optimal regret bounds under episodic and finite-horizon MDPs. Compared to prior results, our regret bounds feature exponential improvements with respect to the horizon length and risk parameter, removing a factor of $e^{|\beta| H^2}$ from existing bounds (where $H$ is the horizon length). This significantly narrows the gap between upper bounds and an existing lower bound of regret.

In summary, we make the following theoretical contributions in this paper.

1. We consider the exponential Bellman equations, which provide an alternative representation for the reward distribution via its MGF. We show that they are also closely related to risk-sensitive RL, as the derived policy improvement optimizes the objective of RL with the entropic risk measure.
2. Inspired by the exponential Bellman equations, we propose a novel analysis of the Bellman backup procedure for RL algorithms based on the entropic risk measure. It further motivates a novel bonus design called doubly decaying bonus. We then design two model-free risk-sensitive RL algorithms equipped with the novel bonus.
3. The novel analytic framework and bonus design together enable us to prove that the preceding algorithms achieve nearly optimal regret bounds, which improve upon existing ones by an exponential factor in terms of the horizon length and risk sensitivity.

## 2 Related works

Distributional RL has been studied in the line of works [3, 11, 15, 24, 26, 27, 30]. In particular, the work of [15] studies distributional RL from the perspective of the characteristic function of the reward distribution, which they name as characteristic value function. While the characteristic value function is similar to the MGF representation that we consider, it is unclear how policy improvement can be implemented on the Bellman equation of characteristic value function, since characteristic functions are complex functions. On the other hand, MGFs are real-valued functions, and therefore policy improvement can be naturally derived from the exponential Bellman equations. Moreover, we provide sample complexity guarantees for risk-sensitive RL by taking advantage of structural properties of the exponential Bellman equations.

The exponential Bellman equations also allow us to connect distributional RL to the problem of RL with the entropic risk measure. This problem is first proposed by the classical work of [19], and has since inspired a large body of studies [2, 4–9, 12–14, 17, 18, 20, 23, 25, 28, 29, 31–33]. However, the algorithms from this line of works require knowledge of the transition kernel or assume access to a simulator of the underlying environment. Theoretical properties of these algorithms are investigated

based on these assumptions, but the results are mostly of asymptotic nature, which do not shed light on their dependency on the parameters of the model and agent.

Recently, the work of [16] investigates the online setting of risk-sensitive RL. Under $K$-episode MDPs with horizon length of $H$, they propose the model-free algorithms RSVI and RSQ, and prove that their algorithms achieve the regret upper bound $\text{Regret}(K) \lesssim e^{|\beta|H^2} \frac{e^{|\beta|H}-1}{|\beta|H}\sqrt{K}$ without assuming knowledge of the transition distribution or access to a simulator. They also provide a lower bound $\text{Regret}(K) \gtrsim \frac{e^{|\beta|H'}-1}{\beta H}\sqrt{K}$ that any algorithm has to incur, where $H'$ is a linear function of $H$. Despite the non-asymptotic nature of their results, it is not hard to see that a wide gap exists between the two bounds. Specifically, the upper bound has an additional $e^{|\beta|H^2}$ factor compared to the lower bound, and even worse, this factor is dominating in the upper bound since the quadratic exponent in $e^{|\beta|H^2}$ makes it exponentially large than $\frac{e^{|\beta|H}-1}{|\beta|H}$ even for moderate values of $|\beta|$ and $H$. Motivated by this gap, our work exploits the exponential Bellman equations to develop a novel analytic framework and bonus design, which lead to algorithms that remove the $e^{|\beta|H^2}$ factor in their regret bounds and significantly narrow the gap between the upper and lower bounds.

Finally, we would like to mention that although exponential Bellman equations have been previously investigated in the risk-sensitive RL literature [2, 5], this is the first time that they are used for deriving regret and sample complexity guarantees of risk-sensitive RL algorithms as well as for connecting distributional and risk-sensitive RL .

**Notations.** For a positive integer $n$, we let $[n] \coloneqq \{1, 2, \ldots, n\}$. For two non-negative sequences $\{a_i\}$ and $\{b_i\}$, we write $a_i \lesssim b_i$ if there exists a universal constant $C > 0$ such that $a_i \leq C b_i$ for all $i$, and write $a_i \asymp b_i$ if $a_i \lesssim b_i$ and $b_i \lesssim a_i$. We use $\tilde{O}(\cdot)$ to denote $O(\cdot)$ while hiding logarithmic factors. For functions $f, g : \mathcal{U} \to \mathbb{R}$, where $\mathcal{U}$ denotes their domain, we write $f \geq g$ if $f(u) \geq g(u)$ for any $u \in \mathcal{U}$. We denote by $\mathbb{I}\{\cdot\}$ the indicator function.

# 3 Problem background

## 3.1 Episodic and finite-horizon MDP

The setting of episodic Markov decision processes can be denoted by $\text{MDP}(\mathcal{S}, \mathcal{A}, H, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $H \in \mathbb{Z}_{>0}$ is the length of each episode, and $\mathcal{P} = \{P_h\}_{h\in[H]}$ and $\mathcal{R} = \{r_h\}_{h\in[H]}$ are the sets of transition kernels and reward functions, respectively. We let $S \coloneqq |\mathcal{S}|$ and $A \coloneqq |\mathcal{A}|$, and we assume $S, A < \infty$. We let $P_h(\cdot \mid s, a)$ denote the probability distribution over successor states of step $h+1$ if action $a$ is executed in state $s$ at step $h$. We assume that the reward function $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is deterministic. We also assume that both $\mathcal{P}$ and $\mathcal{R}$ are unknown to learning agents.

Under the setting of an episodic MDP, the agent aims to learn the optimal policy by interacting with the environment throughout $K > 0$ episodes, described as follows. At the beginning of episode $k$, an initial state $s_1^k$ is selected by the environment. In each step $h \in [H]$ of episode $k$, the agent observes state $s_h^k \in \mathcal{S}$, executes an action $a_h^k \in \mathcal{A}$, and receives a reward equal to $r_h(s_h^k, a_h^k)$ from the environment. The MDP then transitions into state $s_{h+1}^k$ randomly drawn from the transition kernel $P_h(\cdot \mid s_h^k, a_h^k)$. The episode terminates at step $H+1$, in which the agent does not take actions or receive rewards. We define a policy $\pi = \{\pi_h\}_{h\in[H]}$ as a collection of functions $\pi_h : \mathcal{S} \to \mathcal{A}$, where $\pi_h(s)$ is the action that the agent takes in state $s$ at step $h$ of the episode.

## 3.2 Distributional RL

Distributional RL is built upon the following key equation, namely the distributional Bellman equation:

$$\forall h \in [H], \quad Z_h^\pi(s, a) \overset{d}{=} R_h(s, a) + Z_{h+1}^\pi(X', U'), \tag{2}$$

for a fixed policy $\pi$, where $X' \sim P_h(\cdot \mid s, a)$, $U' \sim \pi(\cdot \mid X')$ and $R_h$ is the reward distribution in step $h$. Here, we use $\overset{d}{=}$ to denote equality in distribution. It can be seen that $Z_h^\pi(s, a)$ is the distribution of cumulative rewards under policy $\pi$ at step $h$, when the state and action $(s, a)$ are visited in step $h$.

Based on Equation (2), a distributional Bellman optimality operator is given by

$$[\mathbb{T}Z](s,a) \overset{d}{:=} R_h(s,a) + Z_{h+1}(X', \operatorname*{argmax}_{a' \in \mathcal{A}} \mathbb{E}[Z_{h+1}(X', a')]), \tag{3}$$

where again $X' \sim P_h(\cdot \mid s,a)$. Note that in Equation (3), the optimal action is greedy with respect to the *expectation* of the distribution $Z_{h+1}$. Since risk-sensitive RL often concerns characteristics of the reward distribution beyond its expectation, it is unclear how policy improvement of risk-sensitive policies can be derived directly from the distributional Bellman equation (2). For that reason, most existing distributional RL algorithms resort to heuristics for learning risk-sensitive policies, usually based on some transformations of distribution estimates such as quantiles [10, 11] or empirical distribution functions [3, 30].

Motivated by this problem, we consider the exponential Bellman equation, which takes the form

$$\forall (h,\mu) \in [H] \times \mathbb{R}, \quad e^{\mu \cdot Q_h^\pi(s,a)} = \mathbb{E}_{X'}[e^{\mu(r_h(s,a) + V_{h+1}^\pi(X'))}], \tag{4}$$

where $X' \sim P_h(\cdot \mid s,a)$ and $r_h$ is the deterministic reward function by our assumption. This equation associates the current and future cumulative utilities ($Q_h^\pi$ and $V_{h+1}^\pi$) in a multiplicative way. An implication of Equation (4) is that one may estimate the MGFs of the reward distribution, e.g., by using the estimator of the form

$$w_h(s,a) = \text{SampAvg}(\{e^{\mu(r_h(s_h,a_h) + V_{h+1}(s_{h+1}))} : (s_h,a_h) = (s,a)\}) \tag{5}$$

given some estimate $V_{h+1}$ and for $(s,a) \in \mathcal{S} \times \mathcal{A}$. Here, we denote by $\text{SampAvg}(\mathcal{X})$ the sample average computed over elements in the set $\mathcal{X}$ throughout past episodes, and it can be seen as an empirical MGF of cumulative rewards from step $h+1$.

More importantly, the exponential Bellman equation (4) naturally allows implementation of policy improvement for risk-sensitive policies, in contrast with the distributional Bellman equation (2). We explain this point below as we establish a connection between Equation (4) and the problem of RL with the entropic risk measure.

### 3.3 RL with the entropic risk measure

For each $h \in [H]$, we define the value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$ of a policy $\pi$ as the cumulative utility of the agent at state $s$ of step $h$ under the entropic risk measure, assuming that the agent commits to policy $\pi$ in later steps. Specifically, we define

$$\forall (h,s) \in [H] \times \mathcal{S}, \quad V_h^\pi(s) := \frac{1}{\beta} \log \left\{ \mathbb{E}\left[ e^{\beta \sum_{i=h}^H r_i(s_i, \pi_i(s_i))} \;\middle|\; s_h = s \right] \right\}, \tag{6}$$

where $\beta \neq 0$ is a given risk parameter. The agent aims to maximize his cumulative utility in step 1, that is, to find a policy $\pi$ such that $V_1^\pi(s)$ is maximized for all state $s \in \mathcal{S}$. Under this setting, if $\beta > 0$, the agent is risk-seeking and if $\beta < 0$, the agent is risk-averse. Furthermore, as $\beta \to 0$ the agent tends to be risk-neutral and $V_h^\pi(s)$ tends to the classical value function.

We may also define the action-value function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which is the cumulative utility of the agent who follows policy $\pi$, conditional on a particular state-action pair; formally, this is given by

$$\forall (h,s,a) \in [H] \times \mathcal{S} \times \mathcal{A}, \quad Q_h^\pi(s,a) := \frac{1}{\beta} \log \left\{ \mathbb{E}\left[ e^{\beta \sum_{i=h}^H r_i(s_i,a_i)} \;\middle|\; s_h = s, a_h = a \right] \right\},$$

Under some mild regularity conditions, there always exists an optimal policy, which we denote as $\pi^*$, that yields the optimal value $V_h^*(s) := \sup_\pi V_h^\pi(s)$ for all $(h,s) \in [H] \times \mathcal{S}$ [2]. For all $(s,a) \in \mathcal{S} \times \mathcal{A}$, the Bellman equation associated with a policy $\pi$ is given by

$$\begin{aligned} Q_h^\pi(s,a) &= r_h(s,a) + \frac{1}{\beta} \log \left\{ \mathbb{E}_{s' \sim P_h(\cdot \mid s,a)} \left[ e^{\beta \cdot V_{h+1}^\pi(s')} \right] \right\}, \\ V_h^\pi(s) &= Q_h^\pi(s, \pi(s)), \qquad V_{H+1}^\pi(s) = 0 \end{aligned} \tag{7}$$

for $h \in [H]$. In Equation (7), it can be seen that the action value $Q_h^\pi$ of step $h$ is a non-linear function of the value function $V_{h+1}^\pi$ of the later step. This is in contrast with the linear Bellman equations in

the risk-neutral setting ($\beta \to 0$), where $Q_h^\pi(s,a) = r_h(s,a) + \mathbb{E}_{s'}[V_{h+1}^\pi(s')]$. Based on Equation (7), for $h \in [H]$, the Bellman optimality equation is given by

$$Q_h^*(s,a) = r_h(s,a) + \frac{1}{\beta}\log\left\{\mathbb{E}_{s'\sim P_h(\cdot\,|\,s,a)}\left[e^{\beta\cdot V_{h+1}^*(s')}\right]\right\}, \tag{8}$$
$$V_h^*(s) = \max_{a\in\mathcal{A}} Q_h^*(s,a), \qquad V_{H+1}^*(s) = 0.$$

Given Equations (7) and (8), associating the problem of RL under the entropic risk measure with exponential Bellman equations is rather straightforward: indeed, by taking exponential on both sides of the Equation (7), we get

$$e^{\beta\cdot Q_h^\pi(s,a)} = \mathbb{E}_{s'\sim P_h(\cdot\,|\,s,a)}\left[e^{\beta(r_h(s,a)+V_{h+1}^\pi(s'))}\right], \tag{9}$$

which assumes the same form as the exponential Bellman equation (4). Similarly, an exponential Bellman optimality equation can be derived by taking exponential on both sides of Equation (8), so that

$$e^{\beta\cdot Q_h^*(s,a)} = \mathbb{E}_{s'\sim P_h(\cdot\,|\,s,a)}\left[e^{\beta(r_h(s,a)+V_{h+1}^*(s'))}\right]. \tag{10}$$

This suggests a policy improvement procedure for a risk-sensitive policy $\pi$:

$$\pi_h(s) \leftarrow \operatorname*{argmax}_{a'\in\mathcal{A}} Q_h(s,a') = \begin{cases} \text{argmax}_{a'\in\mathcal{A}}\, e^{\beta\cdot Q_h(s,a')}, & \text{if } \beta > 0 \\ \text{argmin}_{a'\in\mathcal{A}}\, e^{\beta\cdot Q_h(s,a')}, & \text{if } \beta < 0, \end{cases} \tag{11}$$

where $Q_h$ denotes some estimated action-value function.

In the next section, we will discuss how the exponential Bellman equations (9) and (10) inspire the development of a novel analytic framework for risk-sensitive RL. Before proceeding, we introduce a performance metric for the agent. For each episode $k$, recall that $s_1^k$ is the initial state chosen by the environment and let $\pi^k$ be the policy of the agent at the beginning of episode $k$. Then the difference $V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k)$ is called the *regret* of the agent in episode $k$. Therefore, after $K$ episodes, the total regret for the agent is given by

$$\text{Regret}(K) := \sum_{k\in[K]} [V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k)], \tag{12}$$

which serves as the key performance metric studied in this paper.

# 4 Analysis of risk-sensitive RL

## 4.1 Mechanism of existing analysis

In this section, we provide an informal overview of the mechanism underlying the existing analysis of risk-sensitive RL. Let us focus on the case $\beta > 0$ for simplicity of exposition; similar reasoning holds for $\beta < 0$. A key step in the existing regret analysis of RL algorithms is to establish a recursion on the difference $V_h^k - V_h^{\pi^k}$ over $h \in [H]$, where $V_h^k$ is the iterate of an algorithm in step $h$ of episode $k$ and $V_h^{\pi^k}$ is the value function of the policy used in episode $k$. Such approach can be commonly found in the literature of algorithms that use the upper confidence bound [21, 22], in which the recursion takes the form of

$$V_h^k - V_h^{\pi^k} \le V_{h+1}^k - V_{h+1}^{\pi^k} + \psi_h^k, \tag{13}$$

for $\beta \to 0$ and some quantity $\psi_h^k$. The work of [16], which studies the risk-sensitive setting under the entropic risk measure, also follows this approach and derives regret bounds by establishing the recursion of the form

$$V_h^k - V_h^{\pi^k} \le e^{\beta H}\left(V_{h+1}^k - V_{h+1}^{\pi^k}\right) + \frac{1}{\beta}\tilde{b}_h^k + e^{\beta H}\tilde{m}_h^k, \tag{14}$$

where $\tilde{b}_h^k$ denotes the bonus which enforces the upper confidence bound and leads to the inequality $V_h^k \ge V_h^\pi$ for any policy $\pi$, and $\tilde{m}_h^k$ is part of a martingale difference sequence. The derivation of Equation (14) is based on the Bellman equation (7), which shows that the action value $Q_h^{\pi^k}$ is the

sum of the reward $r_h$ and the entropic risk measure of $V_{h+1}^{\pi^k}$. Following [16], we may then unroll the recursion (14) from $h = H$ to $h = 1$ to get

$$V_1^k - V_1^{\pi^k} \leq \frac{1}{\beta} e^{\beta H^2} \sum_h \tilde{b}_h^k + e^{\beta H^2} \sum_h \tilde{m}_h^k, \tag{15}$$

given that $V_{H+1}^k = V_{H+1}^{\pi^k} = 0$. Using the inequality $\text{Regret}(K) \leq \sum_k (V_1^k - V_1^{\pi^k})$, $\sum_{k,h} \tilde{b}_h^k \lesssim (e^{\beta H} - 1)\sqrt{K}$ and $\sum_{k,h} \tilde{m}_h^k \lesssim \sqrt{K}$, we obtain the regret bound in [16] as $\text{Regret}(K) \lesssim e^{\beta H^2} \frac{e^{\beta H} - 1}{\beta H} \sqrt{K}$. Therefore, it can be seen that the dominating factor $e^{\beta H^2}$ in their regret bound originates in Equation (15), which can be further traced back to the exponential factor $e^{\beta H}$ in the error dynamics (14).

### 4.2 Refined approach via exponential Bellman equations

While the existing analysis in (14) is motivated by the Bellman equation of the form given in (7), we propose to work on the exponential Bellman equation (9). Equation (9) operates on the quantities $e^{\beta \cdot Q_h^\pi}$ and $e^{\beta \cdot V_{h+1}^\pi}$, which can be thought of as the MGFs of the current and future values, while the reward function $r_h$ is involved as a multiplicative term. This motivates us to derive a new recursion:

$$e^{\beta \cdot V_h^k} - e^{\beta \cdot V_h^{\pi^k}} \leq e^{\beta \cdot r_h^k} \left( e^{\beta \cdot V_{h+1}^k} - e^{\beta \cdot V_{h+1}^{\pi^k}} \right) + b_h^k + m_h^k, \tag{16}$$

where $b_h^k, m_h^k$ denote some bonus and martingale terms, respectively, and $r_h^k$ stands for the reward in step $h$ of episode $k$. Unrolling Equation (16) yields

$$e^{\beta \cdot V_1^k} - e^{\beta \cdot V_1^{\pi^k}} \leq \sum_h e^{\beta \cdot D_h^k} (b_h^k + m_h^k), \tag{17}$$

where $D_h^k = \sum_{i \in [h-1]} r_i^k$. In words, the error of $e^{\beta \cdot V_1^k} - e^{\beta \cdot V_1^{\pi^k}}$ is bounded by the weighted sum of bonus and martingale difference terms, where the weights are given by $e^{\beta \cdot D_h^k}$, the exponential rewards up to step $h - 1$. We may then apply a localized linearization of the logarithmic function, which gives $\text{Regret}(K) \leq \frac{1}{\beta} \sum_k (e^{\beta \cdot V_1^k} - e^{\beta \cdot V_1^{\pi^k}})$, and arrives at a regret upper bound (the formal regret bounds will be established in Theorems 1 and 2 below). Differerent from Equation (14) where rewards are only implicitly encoded in $V_h^k$, in Equation (16) rewards are explicitly involved in the error dynamics via an exponential term.

To see why Equation (16) is intuitively correct, we may divide both sides of the equation by $\beta$ and take $\beta \to 0$. By doing so, we should expect to obtain quantities from the error dynamics (13) of risk-neutral RL. Since the function $f_\beta(x) = (e^{\beta x} - 1)/\beta$ satisfies that $f_\beta(x) \to x$ as $\beta \to 0$ for any fixed $x$, we have

$$\lim_{\beta \to 0} \frac{1}{\beta} (e^{\beta \cdot V_h^k} - e^{\beta \cdot V_h^{\pi^k}}) = V_h^k - V_h^{\pi^k},$$

$$\lim_{\beta \to 0} \frac{1}{\beta} (e^{\beta \cdot r_h^k} (e^{\beta \cdot V_{h+1}^k} - e^{\beta \cdot V_{h+1}^{\pi^k}})) = r_h^k + V_{h+1}^k - (r_h^k + V_{h+1}^{\pi^k}) = V_{h+1}^k - V_{h+1}^{\pi^k},$$

recovering terms in (13). Therefore, the recursion (16) can be seen as generalizing those in the analysis of risk-neutral RL.

By comparing Equations (16) and (14), we see that while both error dynamics are derived from the same underlying Bellman equation, they inspire drastically different forms of recursion. Note that the multiplicative factor $e^{\beta \cdot r_h^k}$ in Equation (16) is milder than the factor $e^{\beta H}$ in Equation (14), since $r_h^k \in [0, 1]$. This is the source of an improvement of our refined analysis over existing works. On the other hand, the success of applying the error dynamics (16) in our analysis crucially depends on the choice of bonus terms $\{b_h^k\}$, as an improper choice would blow up the error $e^{\beta \cdot V_1^k} - e^{\beta \cdot V_1^{\pi^k}}$. This observation motivates our novel bonus design, as we explain next in Section 5.

**Algorithm 1** RSV2

1: $Q_h(\cdot,\cdot), V_h(\cdot) \leftarrow H-h+1$, $N_h(\cdot,\cdot) \leftarrow 0$ and $w_h(\cdot,\cdot) \leftarrow 0$ for all $h \in [H+1]$
2: **for** episode $k = 1, \ldots, K$ **do**
3:  **for** step $h = H, \ldots, 1$ **do**
4:  **for** $(s,a) \in \mathcal{S} \times \mathcal{A}$ such that $N_h(s,a) \geq 1$ **do**
5:  $w_h(s,a) \leftarrow \frac{1}{N_h(s,a)} \sum_{\tau \in [k-1]} \mathbb{I}\{(s_h^\tau, a_h^\tau) = (s,a)\} \cdot e^{\beta[r_h(s,a)+V_{h+1}(s_{h+1}^\tau)]}$
6:  $b_h(s,a) \leftarrow c|e^{\beta(H-h+1)} - 1|\sqrt{\frac{S\log(HSAK/\delta)}{N_h(s,a)}}$ where $c > 0$ is a universal constant
7:  $G_h(s,a) \leftarrow \begin{cases} \min\{w_h(s,a) + b_h(s,a),\ e^{\beta(H-h+1)}\}, & \text{if } \beta > 0 \\ \max\{w_h(s,a) - b_h(s,a),\ e^{\beta(H-h+1)}\}, & \text{if } \beta < 0 \end{cases}$
8:  $V_h(s) \leftarrow \max_{a' \in \mathcal{A}} \frac{1}{\beta} G_h(s,a')$
9:  **end for**
10:  **end for**
11:  $\forall h \in [H]$, take $a_h \leftarrow \text{argmax}_{a' \in \mathcal{A}} \frac{1}{\beta} G_h(s_h, a')$; observe $r_h(s_h, a_h)$, $s_{h+1}$
12:  Add 1 to $N_h(s_h, a_h)$
13: **end for**

# 5 Algorithms

## 5.1 Overview of algorithms

In this section, we propose two model-free algorithms for RL with the entropic risk measure. We first present RSV2, which is based on value iteration, in Algorithm 1. The algorithm has two main stages: it first estimates the value function using data accumulated up to episode $k-1$ (Line 3–10) and then executes the estimated policy to collect new trajectory (Line 11). In value function estimation, it computes the weights $w_h$, or the empirical MGF of some estimated cumulative rewards evaluated at $\beta$, which can be seen as a simple moving average over $\tau \in [k-1]$. Therefore, Line 5 functions as a concrete implementation of Equation (5) where the sample average is instantiated as a simple moving average. Then in Line 7, it computes an augmented estimate $G_h$ by combining $w_h$ with a bonus term $b_h$ (defined in Line 6). This is followed by thresholding to put $G_h$ in the proper range. Note that $G_h$ is an optimistic estimator of the quantity $e^{\beta \cdot Q_h^\pi}$ in Equation (9): the construction of $G_h$ is augmented by $b_h$ so that it encourages exploration of rarely visited state-action pairs in future episodes, and thereby follows the principle of Risk-Sensitive Optimism in the Face of Uncertainty [16]. When $\beta < 0$, the bonus is subtracted from $w_h$, since a higher level of optimism corresponds to a *smaller* value of the estimate. In addition, Line 11 follows the reasoning of policy improvement suggested in Equation (11).

Next, we introduce RSQ2 in Algorithm 2, which is based on Q-learning. Similar to Algorithm 1, it consists of value estimation (Line 8–11) and policy execution (Line 6) steps. By combining Lines 9 and 10, we see that Algorithm 2 computes the optimistic estimate $G_h$ as a projection of an exponential moving average of empirical MGFs:

$$G_h(s_h, a_h) \leftarrow \Pi_h\{\text{EMA}(\{e^{\beta[r_h(s_h,a_h)+V_{h+1}(s_{h+1})]}\})\}, \tag{18}$$

where $\Pi_h$ denotes a projection that depends on step $h$. In particular, Line 9 can be interpreted as a computation of empirical MGFs evaluated at $\beta$ and thus a concrete implementation of Equation (5) using an exponential moving average. This is in contrast with the simple moving average update in Algorithm 1.

Although Algorithms 1 and 2 are inspired by RSVI and RSQ of [16], respectively, we note that the main novelty of our algorithms lies in the bonus terms ($b_h$ in Algorithm 1 and $b_{h,t}$ in Algorithm 2), which we call the *doubly decaying* bonus. We discuss this new bonus design in the following.

## 5.2 Doubly decaying bonus

Let us focus on $\beta > 0$ for this discussion. In optimism-based algorithms, the bonus term is used to enforce the upper confidence bound in order to encourage sufficient exploration in uncertain environments. It takes the form of a multiplier times a factor that is inversely proportional to visit

**Algorithm 2** RSQ2

$Q_h(\cdot,\cdot), V_h(\cdot) \leftarrow H-h+1$ if $\beta > 0$; $Q_h(\cdot,\cdot), V_h(\cdot) \leftarrow 0$ otherwise, for all $h \in [H+1]$
$N_h(\cdot,\cdot) \leftarrow 0$ for all $h \in [H]$; $\alpha_u \leftarrow \frac{H+1}{H+u}$ for $u \in \mathbb{Z}$
**for** episode $k = 1, \ldots, K$ **do**
  Receive the initial state $s_1$
  **for** step $h = 1, \ldots, H$ **do**
    Take action $a_h \leftarrow \operatorname{argmax}_{a' \in \mathcal{A}} \frac{1}{\beta} G_h(s_h, a')$ and observe $r_h(s_h, a_h)$ and $s_{h+1}$
    Add 1 to $N_h(s_h, a_h)$; $\quad t \leftarrow N_h(s_h, a_h)$
    $b_{h,t} \leftarrow c|e^{\beta(H-h+1)} - 1|\sqrt{\frac{H\log(HSAK/\delta)}{t}}$ for some universal constant $c > 0$
    $w_h(s_h, a_h) \leftarrow (1-\alpha_t) \cdot G_h(s_h, a_h) + \alpha_t \cdot e^{\beta[r_h(s_h,a_h)+V_{h+1}(s_{h+1})]}$
    $G_h(s_h, a_h) \leftarrow \begin{cases} \min\{w_h(s_h,a_h) + \alpha_t b_{h,t},\ e^{\beta(H-h+1)}\}, & \text{if } \beta > 0 \\ \max\{w_h(s_h,a_h) - \alpha_t b_{h,t},\ e^{\beta(H-h+1)}\}, & \text{if } \beta < 0 \end{cases}$
    $V_h(s_h) \leftarrow \max_{a' \in \mathcal{A}} \frac{1}{\beta} G_h(s_h, a')$
  **end for**
**end for**

counts $\{N_h\}$. Our bonus follows this structure and is given by

$$b_h(s,a) \propto (e^{\beta(H-h+1)} - 1)\sqrt{\frac{1}{N_h(s,a)}}, \tag{19}$$

ignoring factors that do not vary in $(h, s, a)$. In Equation (19), the quantity $e^{\beta(H-h+1)}$ plays the role of the multiplier and $\sqrt{1/N_h(s,a)}$ is the factor that decreases in the visit count. While the component $\sqrt{1/N_h(s,a)}$ is common in bonus terms, our new bonus is designed to shrink its multiplier deterministically and exponentially across the horizon steps, as $e^{\beta(H-h+1)} - 1$ decreases from $e^{\beta H} - 1$ in step $h = 1$ to $e^{\beta} - 1$ in step $h = H$. This is in sharp contrast with the bonus terms typically found in risk-neutral RL algorithms, where the multipliers are kept constant in step $h$ (usually as a constant multiple of $H$). Furthermore, this feature is also in contrast with the bonus design in RSVI and RSQ proposed by [16], whose multiplier is $e^{\beta H} - 1$ and kept fixed along the horizon. Because $b_h$ decays both in the visit count $N_h(s,a)$ (across episodes) and the multiplier $e^{\beta(H-h+1)} - 1$ (across the horizon), we name it as *doubly decaying* bonus. We remark that this is a novel feature of Algorithms 1 and 2, compared to RSVI and RSQ. Let us discuss how this new exploration mechanism is motivated from the error dynamics (17).

**Motivation of exponential decay.** From Equation (17), we see that the error of the iterate is bounded by the sum of weighted bonus terms, where the weights are of the form $e^{\beta \cdot D_h}$ and $D_h \in [0, h-1]$. Choosing $b_h \propto e^{\beta(H-h+1)} - 1$ ensures that the weighted bonus is on the order of $e^{\beta H} - 1$ at maximum. On the other hand, if we use the bonus as in [16], which is proportional to $e^{\beta H} - 1$, then we would end up with a multiplicative factor $e^{2\beta H} - 1$ in regret, which is exponentially larger than $e^{\beta H} - 1$. An alternative way to understanding the exponential decay of our bonus is as follows. At step $h$, the estimated value function is $V_h \in [0, H-h+1]$, which implies $e^{\beta \cdot V_h} \in [1, e^{\beta(H-h+1)}]$. The iterate $G_h$ (of Algorithm 1 or 2) is used to estimate $e^{\beta \cdot Q_h^\pi}$, with its estimation error given by

$$|e^{\beta \cdot Q_h^\pi} - G_h| \approx |e^{\beta \cdot Q_h^\pi} - \widehat{\mathbb{P}}_h e^{\beta(r_h + V_{h+1})}| \le e^{\beta(H-h+1)} - 1,$$

where $\widehat{\mathbb{P}}_h$ denotes an empirical aveage operator over historical data in step $h$. Therefore, the estimation error of $G_h$ shrinks exponentially across the horizon. Since bonus is used to compensate for and dominate the estimation error, the miminial order of $b_h$ required is thus $e^{\beta(H-h+1)} - 1$, which is exactly the multiplier in Equation (19).

As a passing note, we remark that the decaying multiplier is not necessary in risk-neutral RL algorithms, since the estimation error therein satisfies $|Q_h - \widehat{\mathbb{P}}_h(r_h + V_{h+1})| \le H - h + 1$, which is upper bounded by $H$ for all $h \in [H]$. This implies that it suffices to simply set the bonus multiplier as a constant multiple of $H$. In contrast, as we have explained, the estimation error of our algorithms decays exponentially in step $h$, and an adaptive and exponentially decaying bonus is needed.

**Comparison with Bernstein-type bonus.** We also compare our bonus in Equation (19) with the Bernstein-type bonus commonly used to improve sample efficiency of risk-neutral RL algorithms [1, 21]. The Bernstein-type bonus takes the form of

$$\bar{b}_h(s,a) \propto \sqrt{\frac{H + \widehat{\mathrm{Var}}(V_{h+1})}{N_h(s,a)}} + o\left(\sqrt{\frac{1}{N_h(s,a)}}\right), \tag{20}$$

where $\widehat{\mathrm{Var}}(\cdot)$ denotes an empirical variance operator over historical data and $o(\cdot)$ denotes a vanishing term as $N_h(s,a) \to \infty$. Our bonus in Equation (19) is different from the Bernstein-type bonus in Equation (20) in mechanism: our bonus features the multiplier $e^{\beta(H-h+1)} - 1$ which decays exponentially and deterministically over $h \in [H]$, whereas the Bernstein-type bonus uses $\sqrt{H + \widehat{\mathrm{Var}}(V_{h+1})}$ as the multiplier (ignoring the vanishing term). The term $\widehat{\mathrm{Var}}(V_{h+1})$ depends on the trajectory of the learning process. Therefore the multiplier is stochastic and stays on the polynomial order of $H$ across the horizon. Moreover, it is unclear how the multiplier behaves in terms of step $h$.

# 6 Main results

In this section, we present and discuss our main theoretical results for Algorithm 1 and 2.

**Theorem 1.** *For any $\delta \in (0,1]$, with probability at least $1-\delta$ there exists a universal constant $c > 0$ (used in Algorithm 1), such that the regret of Algorithm 1 is bounded by*

$$\mathrm{Regret}(K) \lesssim \frac{e^{|\beta|H} - 1}{|\beta|H} \sqrt{H^4 S^2 A K \log^2(HSAK/\delta)}.$$

**Theorem 2.** *For any $\delta \in (0,1]$, with probability at least $1-\delta$ and when $K$ is sufficiently large, there exists a universal constant $c > 0$ (used in Algorithm 2) such that the regret of Algorithm 2 obeys*

$$\mathrm{Regret}(K) \lesssim \frac{e^{|\beta|H} - 1}{|\beta|H} \sqrt{H^3 S A K \log(HSAK/\delta)}.$$

The proof of the two theorems are provided in Appendices A and B, respectively. Note that the above results generalize those in the literature of risk-neutral RL: when $\beta \to 0$, we recover the same regret bounds of LSVI in [22] and Q-learning in [21].

Let us discuss the connections between our results and those in [16]. The work of [16] proposes two algorithms, RSVI and RSQ, that attain the regret bound

$$\mathrm{Regret}(K) \lesssim e^{|\beta|H^2} \cdot \frac{e^{|\beta|H} - 1}{|\beta|H} \sqrt{K}, \tag{21}$$

ignoring polynomial dependencies on other parameters and logarithmic factors, and a lower bound incurred by any algorithm

$$\mathrm{Regret}(K) \gtrsim \frac{e^{|\beta|H'} - 1}{|\beta|H} \sqrt{K}, \tag{22}$$

where $H'$ is a linear function in $H$. In particular, the proof of the lower bound is based on reducing an hard instance of MDP to a multi-armed bandit. It is a priori unclear whether the extra exponential factor $e^{|\beta|H^2}$ in the upper bound (21) is fundamental in the MDP setting, or is due to suboptimal analysis or algorithmic design. We would like to mention that although one trivial way of avoiding the $e^{|\beta|H^2}$ factor in the upper bound (21) is to use a sufficiently small $|\beta|$ in the algorithms of [16] (e.g., $|\beta| \leq \frac{1}{H^2}$ so that $e^{|\beta|H^2} \lesssim 1$), such a small $|\beta|$ defeats the very purpose of ensuring appropriate degree of risk-sensitivity for the algorithms. Hence, an answer for *all* $\beta \neq 0$ would be desirable.

In view of Theorems 1 and 2, we see that our Algorithms 1 and 2 achieve regret bounds that are exponentially sharper than those of RSVI and RSQ. In particular, our results eliminate the $e^{|\beta|H^2}$ factor from Equation (21) thanks to the novel analysis and doubly decaying bonus in our algorithms, which are inspired by the exponential Bellman equations (9). As a result, our bounds significantly narrow the gap between upper bounds and the lower bound (22).

# References


[1] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272, 2017.

[2] Nicole Bäuerle and Ulrich Rieder. More risk-sensitive Markov decision processes. *Mathematics of Operations Research*, 39(1):105–120, 2014.

[3] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458. PMLR, 2017.

[4] Vivek S. Borkar. A sensitivity formula for risk-sensitive cost and the actor-critic algorithm. *Systems & Control Letters*, 44(5):339–346, 2001.

[5] Vivek S. Borkar. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27(2):294–311, 2002.

[6] Vivek S. Borkar. Learning algorithms for risk-sensitive control. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems–MTNS*, pages 55–60, 2010.

[7] Vivek S. Borkar and Sean P. Meyn. Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research*, 27(1):192–209, 2002.

[8] Rolando Cavazos-Cadena and Daniel Hernández-Hernández. Discounted approximations for risk-sensitive average criteria in Markov decision chains with finite state space. *Mathematics of Operations Research*, 36(1):133–146, 2011.

[9] Stefano P. Coraluppi and Steven I. Marcus. Risk-sensitive, minimax, and mixed risk-neutral/minimax control of Markov decision processes. In *Stochastic Analysis, Control, Optimization and Applications*, pages 21–40. Springer, 1999.

[10] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pages 1096–1105. PMLR, 2018.

[11] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[12] Giovanni B. Di Masi and Lukasz Stettner. Risk-sensitive control of discrete-time Markov processes with infinite horizon. *SIAM Journal on Control and Optimization*, 38(1):61–78, 1999.

[13] Giovanni B. Di Masi and Lukasz Stettner. Infinite horizon risk sensitive control of discrete time Markov processes with small risk. *Systems & Control Letters*, 40(1):15–20, 2000.

[14] Giovanni B. Di Masi and Łukasz Stettner. Infinite horizon risk sensitive control of discrete time Markov processes under minorization property. *SIAM Journal on Control and Optimization*, 46(1):231–252, 2007.

[15] Amir-massoud Farahmand. Value function in frequency domain and the characteristic value iteration algorithm. In *Advances in Neural Information Processing Systems*, 2019.

[16] Yingjie Fei, Zhuoran Yang, Yudong Chen, Zhaoran Wang, and Qiaomin Xie. Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret. In *Advances in Neural Information Processing Systems*, 2020.

[17] Wendell H Fleming and William M McEneaney. Risk-sensitive control on an infinite time horizon. *SIAM Journal on Control and Optimization*, 33(6):1881–1915, 1995.

[18] Daniel Hernández-Hernández and Steven I. Marcus. Risk sensitive control of Markov processes in countable state space. *Systems & Control Letters*, 29(3):147–155, 1996.

[19] Ronald A. Howard and James E. Matheson. Risk-sensitive Markov decision processes. *Management Science*, 18(7):356–369, 1972.

[20] Anna Jaśkiewicz. Average optimality for risk-sensitive control with general state space. *The Annals of Applied Probability*, 17(2):654–675, 2007.

[21] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I. Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

[22] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. *arXiv preprint arXiv:1907.05388*, 2019.

[23] Steven I. Marcus, Emmanual Fernández-Gaucherand, Daniel Hernández-Hernandez, Stefano Coraluppi, and Pedram Fard. Risk sensitive Markov decision processes. In *Systems and Control in the Twenty-first Century*, pages 263–279. Springer, 1997.

[24] Borislav Mavrin, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional reinforcement learning for efficient exploration. In *International Conference on Machine Learning*, pages 4424–4434. PMLR, 2019.

[25] Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine Learning*, 49(2-3):267–290, 2002.

[26] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *International Conference on Machine Learning*, 2010.

[27] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Parametric return density estimation for reinforcement learning. *arXiv preprint arXiv:1203.3497*, 2012.

[28] Takayuki Osogami. Robustness and risk-sensitivity in Markov decision processes. In *Advances in Neural Information Processing Systems*, pages 233–241, 2012.

[29] Stephen D. Patek. On terminating Markov decision processes with a risk-averse objective function. *Automatica*, 37(9):1379–1386, 2001.

[30] Mark Rowland, Marc Bellemare, Will Dabney, Rémi Munos, and Yee Whye Teh. An analysis of categorical distributional reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 29–37. PMLR, 2018.

[31] Yun Shen, Wilhelm Stannat, and Klaus Obermayer. Risk-sensitive Markov control processes. *SIAM Journal on Control and Optimization*, 51(5):3652–3672, 2013.

[32] Yun Shen, Michael J. Tobia, Tobias Sommer, and Klaus Obermayer. Risk-sensitive reinforcement learning. *Neural Computation*, 26(7):1298–1328, 2014.

[33] Peter Whittle. *Risk-sensitive Optimal Control*, volume 20. Wiley New York, 1990.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]