Risk-Sensitive Exponential Actor Critic

Alonso Granados, Jason Pacheco

Department of Computer Science, University of Arizona
{alonsog, pachecoj}@arizona.edu

Abstract

Model-free deep reinforcement learning (RL) algorithms have achieved tremendous success on a range of challenging tasks. However, safety concerns remain when these methods are deployed on real-world applications, necessitating risk-aware agents. A common utility for learning such risk-aware agents is the entropic risk measure, but current policy gradient methods optimizing this measure must perform high-variance and numerically unstable updates. As a result, existing risk-sensitive model-free approaches are limited to simple tasks and tabular settings. In this paper, we provide a comprehensive theoretical justification for policy gradient methods on the entropic risk measure, including on- and off-policy gradient theorems for the stochastic and deterministic policy settings. Motivated by theory, we propose risk-sensitive exponential actor-critic (rsEAC), an off-policy model-free approach that incorporates novel procedures to avoid the explicit representation of exponential value functions and their gradients, and optimizes its policy w.r.t. the entropic risk measure. In this way, we show that rsEAC produces more numerically stable updates compared to existing approaches and reliably learns risk-sensitive policies in challenging risky variants of continuous tasks in MuJoCo.

Code — https://github.com/AlonsoGranados/rsEAC

1 Introduction

Model-free deep reinforcement learning (RL) algorithms have been successful at learning complex behaviors using only interactions with the environment (Mnih et al. 2015; Haarnoja et al. 2018; Gu et al. 2017). However, interacting with real-world applications, such as autonomous driving (Mavrin et al. 2019), robotics (Nass, Belousov, and Peters 2019) and finance (Artzner et al. 1999), can lead to catastrophic events, motivating a need for risk-aware agents in decision-making problems. Risk-sensitive RL aims to learn policies that maximize performance metrics that incorporate a penalty for the variability of the return—either due to intrinsic uncertainty in the transition dynamics or randomness in the reward signal.

Risk-sensitive RL has been studied through various risk criteria: variance-related risk measures (Tamar, Di Castro, and Mannor 2012; La and Ghavamzadeh 2013), reward-volatility risk (Bisi et al. 2020), Gini-deviation (Luo et al. 2023), distortion risk (Dabney et al. 2018), and conditional value at risk (CVaR) (Morimura et al. 2010; Chow and Ghavamzadeh 2014). In this paper, we consider risk-sensitive RL with the entropic risk measure. This metric incorporates risk into its objective via the exponential utility function (Howard and Matheson 1972), resulting in a non-linear Bellman equation. In contrast to the standard RL problem, sample-based estimates of the expectation operation are biased. Therefore, most of the work under this framework has been in the context of MDP control, where transition dynamics are known (García and Fernández 2015). The exponential Bellman equations provide a model-free approach that leads to improved regret bounds for risk-sensitive RL (Fei et al. 2021). In the approximate setting, we can minimize the exponentiated TD error, which accelerates the learning process and produces robust policies to model perturbations (Noorani, Mavridis, and Baras 2023). However, the estimation of these functions is numerically unstable.

Policy gradient algorithms improve their policy by updating it in the direction of the performance gradient. The fundamental result for these methods is the policy gradient theorem (Sutton et al. 1999), which enables estimation of the gradient using only sampled trajectories. Policy gradient algorithms have also been extended to handle for general utilities functions (Zhang et al. 2020) and multi-objective problems (Bai, Agarwal, and Aggarwal 2022). In risk-sensitive RL, a gradient estimate can be derived using the likelihood ratio trick (Nass, Belousov, and Peters 2019). However, this estimate may suffer from high variance because its depends on the full trajectory and is scaled by the exponentiated return, which can be numerically unstable.

We now present our contributions to address these shortcomings. First, we derive risk-sensitive policy gradient theorems for the on-policy setting and approximations for the off-policy setting. These results circumvent the need to estimate a gradient for the full trajectory. Second, we study the instabilities that emerge from learning a critic network that approximates the exponentiated return, and propose a representation that avoids its explicit computation by allowing for computation in log-domain. Additionally, we consider a stabilizing mechanism for the critic gradient, based on mini-batch normalization and clipping. Finally, we present

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

risk-sensitive exponential actor-critic (rsEAC), a practical off-policy algorithm that optimizes the entropic risk measure where the actor is optimized using our risk-sensitive off-policy gradient, and the critic incorporates our stabilizing mechanisms to estimate the entropic risk measure for the current policy. We evaluate our algorithm on complex continuous tasks where risk-aversion can be verified, and demonstrate that our method can learn risk-sensitive policies with high-return.

2 Background

Consider a finite horizon Markov decision process (MDP) which is composed of: a state space $\mathcal{S}$, an action space $\mathcal{A}$, an initial state distribution $p_1(s_1)$, a transition dynamics distribution $p(s_{t+1}|s_t, a_t)$ that satisfies the Markov property $p(s_{t+1}|s_t, a_t, \ldots, s_1, a_1) = p(s_{t+1}|s_t, a_t)$, and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which we abbreviate as $r_t := r(s_t, a_t)$. We assume that the agent does not have access to the transition dynamics and the reward function. The agent selects actions based on a policy denoted by $\pi_\theta(a_t|s_t)$, a conditional distribution over $\mathcal{A}$ parameterized by $\theta \in \mathbb{R}^n$. The distribution over a trajectory $\tau = (s_1, a_1, \ldots, s_T, a_t, s_{T+1})$ for a policy π_θ is given by $p_\pi(\tau) = p(s_1) \prod_{t=1}^{T} p(s_{t+1} \mid s_t, a_t)\pi_\theta(a_t \mid s_t)$. Value functions are defined to be the expected cumulative rewards, $V_{\pi_\theta}(s) = \mathbb{E}_{p_\pi(\tau)}[\sum_{t=1}^{T} r_t | s_1 = s]$ and $Q_{\pi_\theta}(s, a) = \mathbb{E}_{p_\pi(\tau)}[\sum_{t=1}^{T} r_t | s_1 = s, a_1 = a]$. We denote the distribution of state s' after transitioning for t time steps from state s by $p(s \to s', t, \pi)$. We also denote the state distribution by $\rho_\pi(s') := \int_S \sum_{t=1}^{T} p_1(s) p(s \to s', t, \pi) \,\mathrm{d}s$. The standard objective in RL is to find a policy that maximizes expected return $J(\pi_\theta) = \mathbb{E}_{p_\pi(\tau)}[\sum_{t=1}^{T} r_t]$.

2.1 Policy Gradient Algorithms

Policy gradient algorithms update the parameters of their policy in the direction of the performance gradient $\nabla_\theta J(\pi_\theta)$. For stochastic policies π_θ, the gradient can be derived from the policy gradient theorem (Sutton et al. 1999),

$$\nabla_\theta J(\pi_\theta) = \int_S \rho_\pi(s) \int_A \nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s, a) \,\mathrm{d}a \,\mathrm{d}s. \quad (1)$$

Although the state distribution $\rho_\pi(s)$ depends on the policy parameters, the gradient surprisingly does not involve the derivative of the state distribution. The work by Silver et al. (2014) extended this result to deterministic policies μ_θ, yielding an analogous gradient that can be estimated more efficiently than the stochastic policy gradient as it lacks the integral over actions:

$$\nabla_\theta J(\mu_\theta) = \int_S \rho_\mu(s) \nabla_\theta \mu_\theta(s) \nabla_a Q_{\mu_\theta}(s, a)|_{a=\mu_\theta(s)} \,\mathrm{d}s. \quad (2)$$

These two results can be extended to off-policy trajectories sampled by a different behavior policy $b(a \mid s) \neq \pi_\theta(a \mid s)$. The performance objective is modified to be the value function of the target policy V_{π_θ}, weighted by the state distribution of the behavior policy, $J_b(\pi_\theta) = \int_S \rho_b(s) V_{\pi_\theta}(s) \,\mathrm{d}s$. The gradient of this objective can be approximated by an off-policy variant of the policy gradient theorem (Degris, White, and Sutton 2012):

$$\nabla_\theta J_b(\pi_\theta) \approx \int_{\mathcal{S}} \int_{\mathcal{A}} \rho_b(s) \nabla_\theta \pi_\theta(a|s) Q_\pi(s, a) \,\mathrm{d}a \,\mathrm{d}s. \quad (3)$$

This approximation drops a term that depends on $\nabla_\theta Q_{\pi_\theta}(s, a)$, because it can be difficult to estimate in an off-policy fashion. However, this approximation still produces policy improvement when using a tabular representation for the policy (Degris, White, and Sutton 2012).

2.2 Risk-Sensitive Objective

In risk-sensitive RL with the entropic risk measure, we aim to find a policy that maximizes:

$$J^\beta(\pi_\theta) = \frac{1}{\beta} \log \mathbb{E}_{p_\pi(\tau)} \left[e^{\beta \sum_{t=1}^{T} r_t} \right], \quad (4)$$

where $\beta \in \mathbb{R}$ controls risk preference. This objective is closely related to mean-variance RL (Mannor and Tsitsiklis 2011)—it admits the Taylor expansion $\mathbb{E}_{p_\pi(\tau)}[\sum_t r_t] + \frac{\beta}{2} \mathrm{Var}_\pi(\sum_t r_t) + O(\beta^2)$ (Mihatsch and Neuneier 2002). Thus, Eq. (4) reduces to the standard (*risk-neutral*) RL objective for $\beta \to 0$. In addition, $\beta > 0$ induces *risk-seeking* policies and $\beta < 0$ induces *risk-averse* policies. We now define the soft-value function as the entropic risk measure of cumulative rewards,

$$Q^\beta_{\pi_\theta}(s, a) = \frac{1}{\beta} \log \mathop{\mathbb{E}}_{p_\pi(\tau)} \left[e^{\beta \sum_{t=1}^{T} r_t} \mid s_1 = s, a_1 = a \right]. \quad (5)$$

This function is recursively associated via a Bellman-style backup equation:

$$Q^\beta_{\pi_\theta}(s_t, a_t) = r_t + \frac{1}{\beta} \log \mathop{\mathbb{E}}_{p_\pi(\cdot|s_t, a_t)} \left[e^{\beta Q^\beta_{\pi_\theta}(s_{t+1}, a_{t+1})} \right]. \quad (6)$$

The main challenge of risk-sensitive RL lies in its nonlinear Bellman equations, as sample-based estimates of the value function are biased due to the presence of the $\log(\cdot)$ operator. For continuous control, an alternative approach is to compute the policy gradient of entropic risk using the likelihood ratio trick (Nass, Belousov, and Peters 2019):

$$\nabla_\theta J^\beta(\pi_\theta) \propto \frac{1}{\beta} \mathop{\mathbb{E}}_{p_\pi(\tau)} \left[\nabla_\theta \sum_{t=1}^{T} \log \pi_\theta(a_t \mid s_t) e^{\beta \sum_{k=1}^{T} r_k} \right]. \quad (7)$$

Although unbiased, this estimator often suffers from numerical instabilities during learning, as it is computed with respect to full trajectories and is proportional to the exponentiated return.

3 Gradients of the Entropic Risk Measure

As discussed previously, the policy gradient of the entropic risk measure may suffer from high variance. In this section, we address this issue by extending the policy gradient framework to risk-sensitive policies under the entropic risk measure. We first introduce policy gradient theorems for the on-policy setting for both stochastic (Thm. 1) and deterministic

(Thm. 2) policies. We then extend these theorems to the off-policy setting by approximating the gradients and demonstrate policy improvements for these approximations when using a tabular policy representation (Thm. 3). All proofs are provided in Appendix A.

3.1 On-policy Risk-Sensitive Policy Gradients

We begin by defining the exponential twisting of dynamics and policy (Chow et al. 2021) as: $p^*(s'|s,a) \propto p(s'|s,a)e^{\beta V^\beta_{\pi_\theta}(s')}$, and $\pi^*_\theta(a|s) \propto \pi_\theta(a|s)e^{\beta Q^\beta_{\pi_\theta}(s,a)}$, respectively, and the state distribution for dynamics $p^*(s'|s,a)$ and policy $\pi^*_\theta(a|s)$ as $\rho^*_\pi(s)$. We now provide our risk-sensitive policy gradient theorem for stochastic policies.

Theorem 1. *(Risk-Sensitive Stochastic Policy Gradient Theorem). The gradient of $J^\beta(\pi_\theta)$ w.r.t. θ is given by:*

$$\frac{1}{\beta}\int_S \rho^*_\pi(s)\int_A \nabla_\theta \pi_\theta(a|s) e^{\beta(Q^\beta_{\pi_\theta}(s,a)-V^\beta_{\pi_\theta}(s))}\,\mathrm{d}a\,\mathrm{d}s. \quad (8)$$

This gradient has two key differences from its risk-neutral version in Eq. (1). First, it is weighted by the state distribution of exponential twisted dynamics and policy, which tend to be optimistic or pessimistic, depending on β. Second, the action-value function is substituted by $e^{\beta(Q^\beta_{\pi_\theta}(s,a)-V^\beta_{\pi_\theta}(s))}$, which can lead to numerical instabilities during learning due to the exponentiated value functions. For deterministic policies, we analogously define the state distribution $\rho^*_\mu(s)$. We now present our risk-sensitive policy gradient for deterministic policies μ_θ.

Theorem 2. *(Risk-Sensitive Deterministic Policy Gradient Theorem). The gradient of $J^\beta(\mu_\theta)$ w.r.t. θ is given by:*

$$\int_S \rho^*_\mu(s)\nabla_\theta \mu_\theta(s)\nabla_a Q^\beta_{\mu_\theta}(s,a)|_{a=\mu_\theta(s)}\,\mathrm{d}s. \quad (9)$$

In particular, we notice that there is no need for an exponential twisted policy. The deterministic gradient is more convenient for two reasons: first, it avoids the integral over actions resulting in a more efficient approach, and second, it avoids the evaluation of an exponential term, which can be numerically unstable during learning.

3.2 Off-policy Risk-Sensitive Policy Gradients

We now consider methods that optimize a risk-sensitive policy from trajectories generated by an arbitrary behavior policy $b(a \mid s)$. To achieve this, we consider the soft-value functions of a target deterministic policy weighted by the state distribution of the behavior policy as our objective, $J^\beta_b(\mu_\theta) = \int_S \rho_b(s)V^\beta_{\mu_\theta}(s)\,\mathrm{d}s$, where ρ_b is the state distribution under the behavior policy. Following Degris, White, and Sutton (2012), we propose an approximation of the gradient $\nabla_\theta J^\beta_b(\mu_\theta)$:

$$\int_{\mathcal{S}} \rho_b(s)\nabla_\theta \mu_\theta(s)\nabla_a Q^\beta_{\mu_\theta}(s,a)|_{a=\mu_\theta(s)}\,\mathrm{d}s := g(\mu_\theta). \quad (10)$$

This approximation drops a term that depends on $\nabla_\theta Q^\beta_{\mu_\theta}(s,a)$. We show that this approximation results in policy improvement when using a tabular representation for the policy.

Theorem 3. *(Deterministic Policy Improvement). Given a policy μ with parameters θ and a linear tabular representation, let $\theta' = \theta + \alpha g(\mu_\theta)$. Then there exist an ϵ such that for all $\alpha < \epsilon$:*

$$V^\beta_{\mu_\theta}(s) \le V^\beta_{\mu_{\theta'}}(s) \qquad \forall s \in S. \quad (11)$$

For completeness, we consider an equivalent result for an off-policy performance objective that uses a stochastic policy π_θ as its target policy in Appendix A.

4 Risk-Sensitive Exponential Actor-Critic

Motivated by theoretical results established in Sec. 3, we now derive a risk-sensitive off-policy actor-critic algorithm that avoids sampling the entire trajectory. We begin by illustrating the issues that arise when learning a critic network that approximates the exponential value function, using the squared exponential temporal-difference (TD). We then address these problems by introducing a critic representation that avoids explicit computation of the exponential value function, and design a stabilizing mechanism for the critic gradient, based on batch normalization and clipping. Finally, we combine our improvements to propose: risk-sensitive exponential actor-critic (rsEAC), a practical algorithm that considerably improves the stability of the optimization of the entropic risk measure.

4.1 Numerical Issues of Exponential TD Learning

As discussed in Sec. 2.2, sample-based estimates of $Q^\beta_{\mu_\theta}$ are biased due to nonlinearity. To obtain unbiased estimates, we consider the exponential value function:

$$Z^\beta_{\mu_\theta}(s_t,a_t) = e^{\beta Q_{\mu_\theta}(s_t,a_t)}. \quad (12)$$

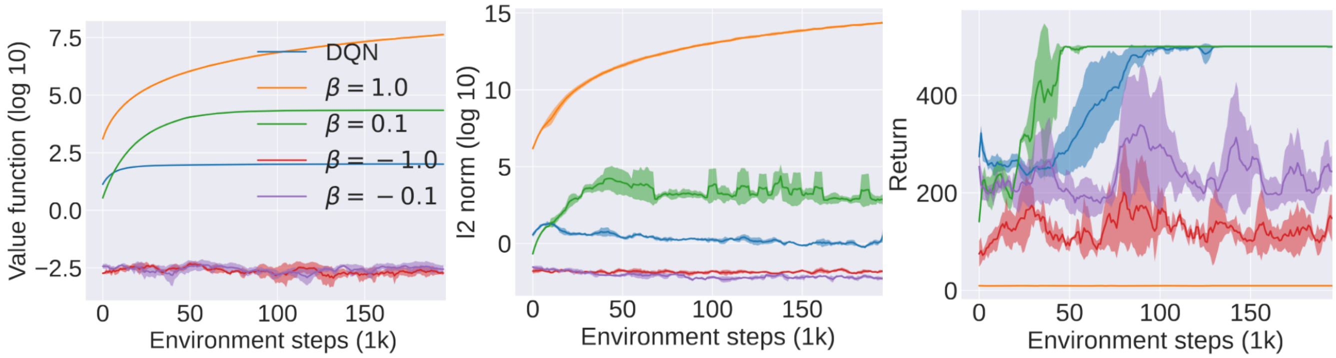
This function follows the exponential Bellman equation, which is obtained by applying an exponential transformation to both sides of Eq. (6):

$$Z^\beta_{\mu_\theta}(s_t,a_t) = e^{\beta r_t}\mathbb{E}_{p_\pi(\cdot|s_t,a_t)}\left[Z^\beta_{\mu_\theta}(s_{t+1},a_{t+1})\right]. \quad (13)$$

Using this equation, $Z^\beta_{\mu_\theta}$ can be estimated using only samples, which can then be used to recover $Q^\beta_{\mu_\theta}$ (Fei et al. 2021). However, it is impractical to learn different exponential value functions for each state and action pair, especially in environments with large state/action spaces. Instead, these functions can be approximated with a parameterized critic network $Z_\psi(s_t,a_t)$, which is trained using stochastic gradient descent to minimize the squared exponential TD error:

$$\mathop{\mathbb{E}}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[\left(Z_\psi(s_t,a_t) - e^{\beta r_t}Z_{\psi'}(s_{t+1})\right)^2\right], \quad (14)$$

where $\mathcal{D}$ is an experience replay buffer that stores previously seen interactions with the environment. The value function $Z_{\psi'}(s_{t+1})$ is implicitly parameterized with a target critic network as $\max_{a_{t+1}} Z_{\psi'}(s_{t+1},a_{t+1})$. Risk-sensitive actor critic (R-AC) (Noorani, Mavridis, and Baras 2023, 2025) considers a similar loss in the context of online learning for its critic. However, two major drawbacks persist in the learning the R-AC critic network: first, the value function estimates the exponentiated reward, $\mathbb{E}[e^{\beta\sum_t r_t}]$, which can lead



(a) Average Initial Value (b) L2 Gradient Norm (c) Average Performance

Figure 1: We train value functions for DQN and $Z_\psi(s_t, a_t)$ for a range of β settings $(-1, -0.1, 0.1, 1)$. We plot the average log-value estimates (*left*), the L2 gradient norm in log-domain (*center*), and the average return (*right*) for 20 episodes over 200k environment steps. For $\beta < 0$, we observe that the value estimates and the gradient norm are almost zero, leading to unstable learning. For $\beta > 0$, the value estimates and gradients are orders of magnitude larger than DQN's estimates. These estimates are used in the gradients, resulting in exploding values.

to numerical overflow / underflow of the network weights. Second, directly minimizing Eq. (14) may cause exploding (or vanishing) gradients, resulting in divergent behavior. Fig. 1 demonstrates these instabilities by learning value functions in the CartPole task (Barto, Sutton, and Anderson 1983) in Gymnasium (Towers et al. 2023).

4.2 Stabilizing Updates

We now consider a critic representation that avoids numerical instabilities by parameterizing the exponential value function as $Z_\psi(s, a) = e^{Q_\psi(s,a)}$ where $Q_\psi(s, a)$ is a neural network. In particular, $\frac{1}{\beta} Q_\psi(s, a)$ serves as our approximation of the soft-value function $Q^\beta_{\mu_\theta}$. This reparameterization is more numerically stable as gradients ∇Q_ψ are taken in the log-domain. We train this function by minimizing the squared exponential TD error $J_Q(\psi)$:

$$\mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[\left(e^{Q_\psi(s_t,a_t)} - e^{\beta r_t + Q_{\psi'}(s_{t+1})}\right)^2\right]. \quad (15)$$

where $\mathcal{D}$ is the experience replay buffer. We minimize Eq. (15) using stochastic gradient descent. The value function $Q_{\psi'}(s_{t+1})$ is also implicitly parameterized with a target critic network as $\max_{a_{t+1}} Q_{\psi'}(s_{t+1}, a_{t+1})$. The gradient $\nabla J_Q(\psi)$ is then:

$$e^{Q_\psi(s_t,a_t)}\left(e^{Q_\psi(s_t,a_t)} - e^{\beta r_t + Q_{\psi'}(s_{t+1})}\right)\nabla Q_\psi(s_t, a_t). \quad (16)$$

Note that the trailing gradient ∇Q_ψ is taken in log-domain, and so is numerically stable. Ignoring this term the leading factors are of the form: $e^x(e^x - e^y)$, which can be numerically unstable. First, we rearrange factors so that the numerical instability occurs only in the leading term. To do this we factor out e^x or e^y, whichever is larger. We define the helper function,

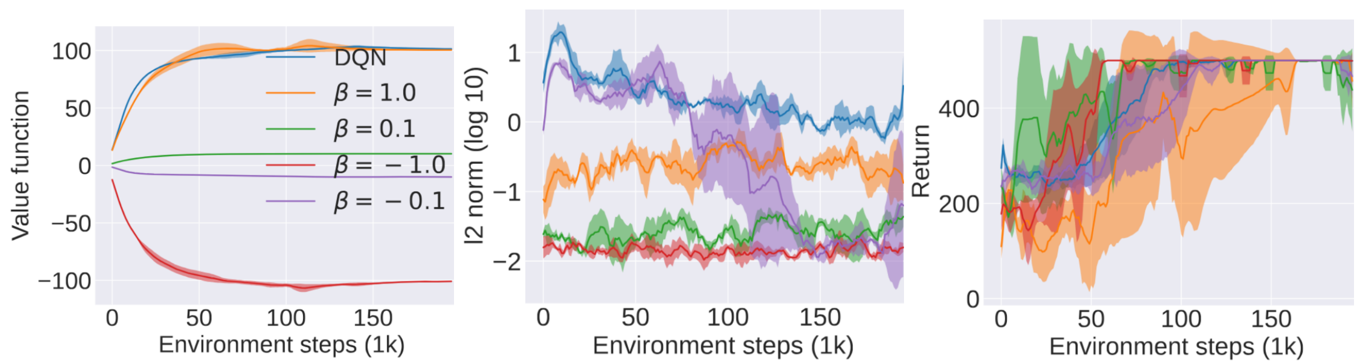
$$f(x, y) \triangleq \begin{cases} (1 - e^{y-x}), & \text{if } x \geq y \\ (e^{x-y} - 1), & \text{otherwise.} \end{cases} \quad (17)$$

Then we have that $e^x(e^x - e^y) = e^{x+\max(x,y)} f(x, y)$. Observe that the trailing term $f(x, y)$ is numerically stable as it is bounded: $f(x, y) \in [-1, 1]$. Numerical instability occurs only in the leading term $e^{x+\max(x,y)}$, which must be stabilized by subtracting a constant z in log-domain, which is equivalent to normalizing $e^x(e^x - e^y) \propto e^{x+\max(x,y)-z} f(x, y)$. We defer the details of our normalizing choice z and our clipping mechanism to Appendix C.

Evaluation We now show that using the normalized clipped gradient addresses the numerical issues of $Z_\psi(s_t, a_t)$ and can stabilize the learning of the exponential-TD objective. Again we consider the CartPole environment, and train the value functions $Q_\psi(s_t, a_t)$ for the same set of β values as in Fig. 1. In Fig. 2a, we observe that the value function $Q_\psi(s_t, a_t)$ is less prone to suffer from numerical overflow compared to $Z_\psi(s_t, a_t)$. In fact, the magnitude learned by DQN (Mnih et al. 2015) is close to $\frac{1}{\beta} Q_\psi(s_t, a_t)$, which is not surprising given that CartPole is a deterministic task and our approach scales the rewards by β. Using the normalized clipped gradient, our approach also guarantees gradients with more manageable magnitudes compared to directly optimizing the exponential-TD objective (Fig. 2b). Finally, our algorithm is capable of learning optimal policies for the all the tested β initializations, which was only possible for one setting when learning $Z_\psi(s_t, a_t)$ (See Fig. 2c).

4.3 Actor-Critic Optimization

We now present risk-sensitive exponential actor-critic (rsEAC), a practical off-policy algorithm that optimizes the entropic risk-measure. We develop an actor-critic algorithm that updates its policy in the direction of the off-policy deterministic policy gradient in Eq. (10), and substitutes the soft-value function $Q^\beta_{\mu_\theta}$ with our approximation Q_ψ. We build on the twin delayed deep deterministic policy gradient algorithm (TD3) (Fujimoto, Hoof, and Meger 2018) for its mechanisms on over-estimation bias reduction and its em-



(a) Average Initial Value (b) L2 Gradient Norm (c) Average Performance

Figure 2: We train value functions for DQN and $Q_\psi(s_t, a_t)$ using the normalized clipped gradient for a range of β settings $(-1, -0.1, 0.1, 1)$. We plot the average value estimates (*left*), the L2 gradient norm in log-domain (*center*), and the average return (*right*) for 20 episodes over 200k environment steps.

pirical success in continuous control tasks. Similarly to TD3, we maintain a pair of critics with parameters ψ_1 and ψ_2, a single actor with parameter θ, and their respective target networks ψ_1', ψ_2' and θ'. For each timestep, we update the pair of critic networks by replacing the target in the normalized clipped gradient derived in Section 4.2 with

$$y_t = \begin{cases} \beta r_t + \min_{i=1,2} \gamma Q_{\psi_i'}(s_{t+1}, a_{t+1}), & \text{if } \beta > 0 \\ \beta r_t + \max_{i=1,2} \gamma Q_{\psi_i'}(s_{t+1}, a_{t+1}), & \text{if } \beta < 0. \end{cases} \quad (18)$$

where $a_{t+1} = \mu_{\theta'}(s_{t+1}) + \epsilon$ and $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma^2), -c^*, c^*)$. We also include a discount factor γ to improve stability for long-horizon tasks. The actor is a deterministic policy parameterized with a neural network $\mu_\theta(s_t)$ and is optimized w.r.t. the risk-sensitive off-policy gradient in Eq. (10), for which we substitute the soft-value function with our estimate $\frac{1}{\beta} Q_{\psi_1}(s_t, a_t)$:

$$\mathbb{E}_{s_t \sim \mathcal{D}} \left[\frac{1}{\beta} \nabla_\theta \mu_\theta(s_t) \nabla_{a_t} Q_{\psi_1}(s_t, a_t) \mid_{a_t = \mu_\theta(s_t)} \right], \quad (19)$$

where s_t is sampled from the buffer $\mathcal{D}$. Given that the policy $\mu_\theta(s_t)$ is deterministic, we ensure adequate exploration by adding Gaussian noise. We update the target networks using an exponentially moving average of the weights. Pseudocode for rsEAC can be found in Appendix E.

5 Related Work

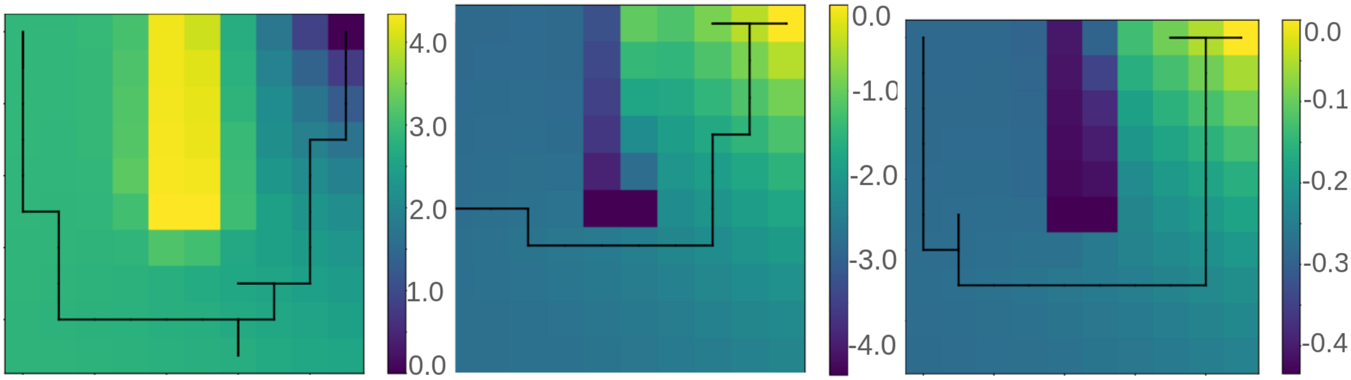
Entropic risk was first investigated by Howard and Matheson (1972) as an approach to incorporate risk sensitivity in MDPs. Following this work, risk-sensitive MDPs—where the agent knows the transition dynamics—have been extensively studied (Fleming and McEneaney 1995; Hernández-Hernández and Marcus 1996; Coraluppi and Marcus 1999; Di Masi and Stettner 1999; Borkar and Meyn 2002; Huang and Haskell 2020). More recently, risk-sensitive RL has been connected to the RL-as-inference framework (Noorani and Baras 2022), in which policy search is formulated as maximum likelihood estimation (Todorov 2008; Kappen, Gómez, and Opper 2012; Levine 2018). The entropic risk measure can also be expressed as an expectation under an auxiliary occupation measure penalized by a KL divergence term (Noorani, Mavridis, and Baras 2025), where policy search is cast as an expectation-maximization style algorithm (Neumann 2011; Abdolmaleki et al. 2018; Chow et al. 2021; Granados, Ebrahimi, and Pacheco 2025). Similarly, trust region policy optimization methods (Schulman et al. 2015; Nachum et al. 2018) employ a KL regularizer; however, its purpose is fundamentally different—it enforces a trust region for stability rather than risk-sensitivity.

Risk-sensitive model-free algorithms optimize the exponential criteria as an equivalent objective of the entropic risk measure (Bäuerle and Rieder 2014; Borkar 2001; Enders, Harrison, and Schiffer 2024). These approaches employ exponential value functions—functions that are associated in a multiplicative way, rather than in an additive way as in standard RL (Fei et al. 2021)—that can be estimated using Monte Carlo methods. However, the exponential criteria suffer from numerical instabilities that arise from the estimation of the expected exponentiated return. Risk-sensitive policy gradient algorithms (Nass, Belousov, and Peters 2019) may suffer from these same instabilities.

Network saturation caused by the exponential function is a well-known issue in neural network design (Goodfellow 2016). While log-likelihood objectives can mitigate this by inverting the exponential, they are incompatible with exponential TD learning, which requires a squared loss. Motivated by batch normalization (Ioffe and Szegedy 2015), we design a normalization scheme for the last layer to avoid gradient saturation. We further stabilize training with gradient clipping on the exponent, reducing exploding and vanishing gradients. To our knowledge, this is the first actor–critic algorithm to incorporate explicit stabilization mechanisms for exponential TD learning, supported by policy gradient theorems that provide a rigorous framework.

6 Experiments

Our experiments aim to: verify that numerical instabilities are a direct result of optimizing the exponential Bell-



(a) Risk-averse ($\beta = -1$) (b) Risk-Seeking ($\beta = 1$) (c) Risk-neutral ($\beta = 0.1$)

Figure 3: Optimal value functions $Z^*(s_t)$ (in log-domain) and corresponding trajectory for different β values. Agents trained with negative β values learn risk-averse policies that avoid the cliff-region, while agents trained with positive β values tend to be risk-seeking and hug the cliff closely. We also recover risk-neutral policies when the magnitude of $|\beta|$ is small.

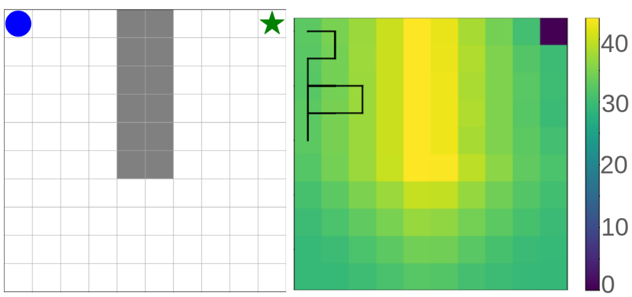


Figure 4: *Left*: 2D grid environment with initial-state and goal-state given by blue circle and green star, respectively. The cliff region is given by gray states. *Right*: We plot the learned optimal value functions (in log-domain) for $\beta = -10$ and a sampled trajectory. The estimated values explode in magnitude, resulting in a policy incapable of reaching the destination.

man equation, analyze the risk preference and robustness of rsEAC to different values of β, and demonstrate that our approach produces competitive risk-sensitive policies compared to existing methods. We begin with a tabular environment to study the objective without additional approximation error, then we compare rsEAC to risk-averse baselines on three challenging risky variations of MuJoCo tasks (Todorov, Erez, and Tassa 2012), where effective risk-aware agents must avoid noisy regions. See Appendix B for additional experiments, in which we show that rsEAC learns stable policies as the risk parameter β varies.

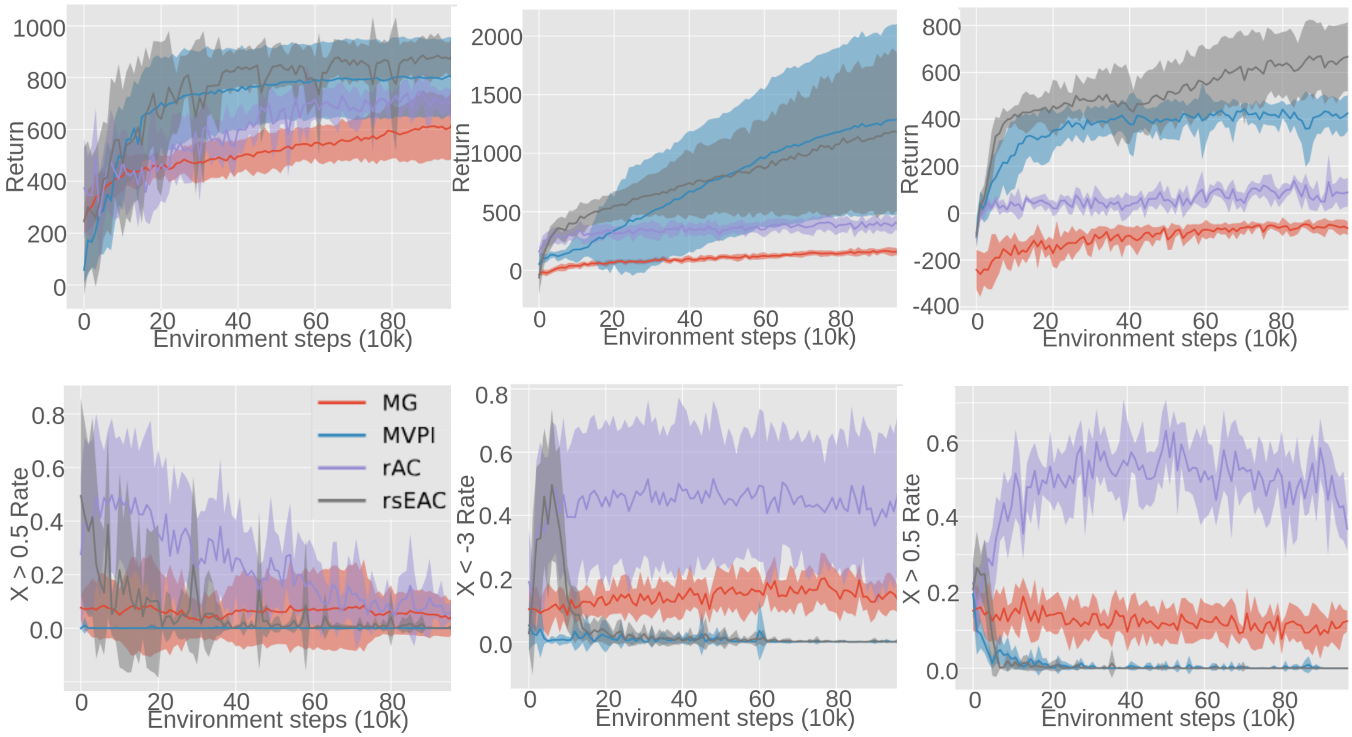
6.1 GridWorld Environment

In this experiment, we illustrate how β modulates risk-sensitivity and identify potential numerical instabilities that may arise when optimizing the exponential Bellman equation in the tabular setting. We use the 2D GridWorld from (Eysenbach et al. 2022), where the agent starts in an initial position and aims to reach a specified goal. At each step, the agent chooses an action (up, left, down and right) but moves in the chosen direction or to a random neighboring state with probability 0.2. To introduce aleatoric risk, we define a cliff-region that the agent must avoid—falling into the cliff yields a large negative reward and terminates the episode. The GridWorld consists of a 10×10 grid (see Fig. 4), where the initial state and goal are marked by a blue circle and green star, respectively, and the cliff by a gray rectangle. Each move incurs a cost of -1 and falling into the cliff results in a penalty of -10. We learn the value functions $Z^*(s_t, a_t)$ using the exponential Bellman equation with an ϵ-greedy policy ($\epsilon = 0.1$) and a discount factor of $\gamma = 0.85$. The algorithm is trained for 50,000 episodes.

In Fig. 3, we plot learned optimal value functions $Z^*(s_t)$ (in log-domain) and a sampled trajectory using a learned optimal policy π^* for a range of β values. We observe that policies trained with $\beta < 0$ prefer longer paths that avoid the cliff-region (Fig. 3a), while policies trained with $\beta > 0$ choose the shortest path which travels next to the cliff (Fig. 3b). As expected, we recover risk-neutral policies when the magnitude of $|\beta|$ is small (Fig. 3c). The learned value functions also demonstrate exploding/vanishing gradient magnitudes when estimated using $|\beta|$ values with larger magnitudes (Fig. 4). We expect these numerical instabilities to develop even more often when introducing function approximators.

6.2 Simulated Robotic Benchmarks

We evaluate rsEAC on three modified MuJoCo continuous control tasks: Swimmer, HalfCheetah, and Ant. Following Luo et al. (2023), we modify their reward functions to make the speed positive in both directions so the agent is free to move left or right. We also construct risky regions in these tasks based on the X-position. We add a stochastic reward sampled from $\mathcal{N}(0, 10^2)$ for Swimmer and HalfCheetah, when the agent's position is greater than 0.5 and smaller than -3, respectively. For the Ant task, we add a stochastic reward sampled from $\mathcal{N}(0, 7^2)$ when its position is greater than 0.5. For all tasks, we append the X-position into the agent's ob-



(a) Swimmer-v4 (b) HalfCheetah-v4 (c) Ant-v4

Figure 5: *Top row*: Average return on MuJoCo benchmarks with risky regions. *Bottom row*: Percentage of steps on an episode in risky regions. The solid curves correspond to the mean and shaded regions to $\pm$ one standard deviation over 10 random seeds.

servation. Hence, a risk-averse agent would do its best effort to avoid these risky regions.

We compare our algorithm against three risk-averse baselines: Mean Gini deviation (MG) (Luo et al. 2023), a policy gradient algorithm that considers Gini deviation as its risk measure and outperforms other return variance methods; mean-variance policy iteration (MVPI) (Zhang, Liu, and Whiteson 2021), a flexible algorithm that optimizes reward-volatility risk measure and demonstrates great empirical success in risk-averse continuous domains; and risk-sensitive actor critic (R-AC) (Noorani, Mavridis, and Baras 2023), an online actor-critic algorithm that optimizes the entropic risk measure. To make the comparison fair between different methods, we implement every actor-critic algorithm on top of TD3, and use the same network architecture across all algorithms. For MG, we use PPO (Schulman et al. 2017) to learn its policy gradient, as suggested by the author's implementation. See Appendix D for additional details.

We report the average return (*top-row*) and the risky region visiting rate (*bottom-row*) for the three risk-averse MuJoCo environments (See Fig. 5). The results show that rsEAC produces policies that are risk-averse (low rate in risky regions), and perform comparably to other baselines in terms of average return. In particular, it outperforms R-AC in every task, showing that our stabilizing mechanisms play an essential role in the learning of risk-averse and high-return policies. Our approach performs comparably to MVPI in terms of final performance across environments, while outperforming it in the higher-dimensional Ant task.

7 Conclusion

In this paper, we study risk-sensitive reinforcement learning using entropic risk. We establish that existing policy gradient approaches learn high-variance and numerically unstable estimates that may lead to divergent behavior by the agent. To resolve these issues, we derive policy gradient theorems that circumvent the need to estimate a gradient for the full trajectory, and propose a critic parameterization for the exponential value function that avoids its explicit computation. We optimize this critic using stochastic gradient descent, for which we too propose a stabilizing mechanism based on batch normalization and clipping. Taking these components together, we propose risk-sensitive exponential actor-critic (rsEAC), a practical algorithm that greatly improves the stability of optimizing the entropic risk measure. Our evaluations demonstrate that rsEAC can learn risk-sensitive and high-return policies in complex continuous tasks.

Limitations First, there is an inherent instability when working with the exponential function, and even our proposed solutions may not fully stabilize updates in all settings. Second, a key component of our method is the risk parameter β, which must be tuned to balance the amount of risk-sensitivity for different tasks. Finally, our policy-gradient approach does not provide regret guarantees.

Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-22-1-0194.

References

Abdolmaleki, A.; Springenberg, J. T.; Tassa, Y.; Munos, R.; Heess, N.; and Riedmiller, M. 2018. Maximum a Posteriori Policy Optimisation. In *International Conference on Learning Representations*.

Artzner, P.; Delbaen, F.; Eber, J.-M.; and Heath, D. 1999. Coherent measures of risk. *Mathematical finance*, 9(3): 203–228.

Bai, Q.; Agarwal, M.; and Aggarwal, V. 2022. Joint optimization of concave scalarized multi-objective reinforcement learning with policy gradient based algorithm. *Journal of Artificial Intelligence Research*, 74: 1565–1597.

Barto, A. G.; Sutton, R. S.; and Anderson, C. W. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5): 834–846.

Bäuerle, N.; and Rieder, U. 2014. More risk-sensitive Markov decision processes. *Mathematics of Operations Research*, 39(1): 105–120.

Bisi, L.; Sabbioni, L.; Vittori, E.; Papini, M.; and Restelli, M. 2020. Risk-Averse Trust Region Optimization for Reward-Volatility Reduction. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 4583–4589. International Joint Conferences on Artificial Intelligence Organization. Special Track on AI in FinTech.

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

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

Chow, Y.; Cui, B.; Ryu, M.; and Ghavamzadeh, M. 2021. Variational Model-based Policy Optimization. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2292–2299. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Chow, Y.; and Ghavamzadeh, M. 2014. Algorithms for CVaR optimization in MDPs. *Advances in neural information processing systems*, 27.

Coraluppi, S. P.; and Marcus, S. I. 1999. *Risk-sensitive, minimax, and mixed risk-neutral/minimax control of Markov decision processes*. Springer.

Dabney, W.; Ostrovski, G.; Silver, D.; and Munos, R. 2018. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, 1096–1105. PMLR.

Degris, T.; White, M.; and Sutton, R. S. 2012. Off-policy actor-critic. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 179–186.

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

Enders, T.; Harrison, J.; and Schiffer, M. 2024. Risk-sensitive soft actor-critic for robust deep reinforcement learning under distribution shifts. *arXiv preprint arXiv:2402.09992*.

Eysenbach, B.; Khazatsky, A.; Levine, S.; and Salakhutdinov, R. R. 2022. Mismatched no more: Joint model-policy optimization for model-based rl. *Advances in Neural Information Processing Systems*, 35: 23230–23243.

Fei, Y.; Yang, Z.; Chen, Y.; and Wang, Z. 2021. Exponential bellman equation and improved regret bounds for risk-sensitive reinforcement learning. *Advances in neural information processing systems*, 34: 20436–20446.

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

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

García, J.; and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1): 1437–1480.

Goodfellow, I. 2016. Deep learning.

Granados, A.; Ebrahimi, R.; and Pacheco, J. 2025. Risk-Sensitive Variational Actor-Critic: A Model-Based Approach. In *International Conference on Learning Representations*.

Gu, S.; Holly, E.; Lillicrap, T.; and Levine, S. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, 3389–3396. IEEE.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

Hernández-Hernández, D.; and Marcus, S. I. 1996. Risk sensitive control of Markov processes in countable state space. *Systems & control letters*, 29(3): 147–155.

Howard, R. A.; and Matheson, J. E. 1972. Risk-sensitive Markov decision processes. *Management science*, 18(7): 356–369.

Huang, W.; and Haskell, W. B. 2020. Stochastic approximation for risk-aware Markov decision processes. *IEEE Transactions on Automatic Control*, 66(3): 1314–1320.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. pmlr.

Kappen, H. J.; Gómez, V.; and Opper, M. 2012. Optimal control as a graphical model inference problem. *Machine learning*, 87: 159–182.

La, P.; and Ghavamzadeh, M. 2013. Actor-critic algorithms for risk-sensitive MDPs. *Advances in neural information processing systems*, 26.

Levine, S. 2018. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.

Luo, Y.; Liu, G.; Poupart, P.; and Pan, Y. 2023. An alternative to variance: Gini deviation for risk-averse policy gradient. *Advances in Neural Information Processing Systems*, 36: 60922–60946.

Mannor, S.; and Tsitsiklis, J. N. 2011. Mean-variance optimization in Markov decision processes. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 177–184.

Mavrin, B.; Yao, H.; Kong, L.; Wu, K.; and Yu, Y. 2019. Distributional reinforcement learning for efficient exploration. In *International conference on machine learning*, 4424–4434. PMLR.

Mihatsch, O.; and Neuneier, R. 2002. Risk-sensitive reinforcement learning. *Machine learning*, 49: 267–290.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Morimura, T.; Sugiyama, M.; Kashima, H.; Hachiya, H.; and Tanaka, T. 2010. Nonparametric return distribution approximation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 799–806. Citeseer.

Nachum, O.; Norouzi, M.; Xu, K.; and Schuurmans, D. 2018. Trust-PCL: An Off-Policy Trust Region Method for Continuous Control. In *International Conference on Learning Representations*.

Nass, D.; Belousov, B.; and Peters, J. 2019. Entropic risk measure in policy search. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1101–1106. IEEE.

Neumann, G. 2011. Variational inference for policy search in changing situations. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 817–824.

Noorani, E.; and Baras, J. S. 2022. A probabilistic perspective on risk-sensitive reinforcement learning. In *2022 American Control Conference (ACC)*, 2697–2702. IEEE.

Noorani, E.; Mavridis, C. N.; and Baras, J. S. 2023. Exponential TD Learning: A Risk-Sensitive Actor-Critic Reinforcement Learning Algorithm. In *2023 American Control Conference (ACC)*, 4104–4109. IEEE.

Noorani, E.; Mavridis, C. N.; and Baras, J. S. 2025. Risk-sensitive reinforcement learning with exponential criteria. *IEEE Transactions on Cybernetics*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*, 387–395. Pmlr.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Tamar, A.; Di Castro, D.; and Mannor, S. 2012. Policy gradients with variance related risk criteria. In *Proceedings of the twenty-ninth international conference on machine learning*, 387–396.

Todorov, E. 2008. General duality between optimal control and estimation. In *2008 47th IEEE Conference on Decision and Control*, 4286–4292. IEEE.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033. IEEE.

Towers, M.; Terry, J. K.; Kwiatkowski, A.; Balis, J. U.; Cola, G. d.; Deleu, T.; Goulão, M.; Kallinteris, A.; KG, A.; Krimmel, M.; Perez-Vicente, R.; Pierré, A.; Schulhoff, S.; Tai, J. J.; Shen, A. T. J.; and Younis, O. G. 2023. Gymnasium.

Zhang, J.; Koppel, A.; Bedi, A. S.; Szepesvari, C.; and Wang, M. 2020. Variational policy gradient method for reinforcement learning with general utilities. *Advances in Neural Information Processing Systems*, 33: 4572–4583.

Zhang, S.; Liu, B.; and Whiteson, S. 2021. Mean-variance policy iteration for risk-averse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10905–10913.