A Risk-Sensitive Approach to Policy Optimization

Jared Markowitz[1], **Ryan W. Gardner**[1], **Ashley Llorens**[2], **Raman Arora**[3], **I-Jeng Wang**[1]

[1]Johns Hopkins University Applied Physics Laboratory
[2] Microsoft Corporation
[3]Johns Hopkins University
Jared.Markowitz@jhuapl.edu

Abstract

Standard deep reinforcement learning (DRL) aims to maximize expected reward, considering collected experiences equally in formulating a policy. This differs from human decision-making, where gains and losses are valued differently and outlying outcomes are given increased consideration. It also fails to capitalize on opportunities to improve safety and/or performance through the incorporation of distributional context. Several approaches to distributional DRL have been investigated, with one popular strategy being to evaluate the projected distribution of returns for possible actions. We propose a more direct approach whereby risk-sensitive objectives, specified in terms of the cumulative distribution function (CDF) of the distribution of full-episode rewards, are optimized. This approach allows for outcomes to be weighed based on relative quality, can be used for both continuous and discrete action spaces, and may naturally be applied in both constrained and unconstrained settings. We show how to compute an asymptotically consistent estimate of the policy gradient for a broad class of risk-sensitive objectives via sampling, subsequently incorporating variance reduction and regularization measures to facilitate effective on-policy learning. We then demonstrate that the use of moderately "pessimistic" risk profiles, which emphasize scenarios where the agent performs poorly, leads to enhanced exploration and a continual focus on addressing deficiencies. We test the approach using different risk profiles in six OpenAI Safety Gym environments, comparing to state of the art on-policy methods. Without cost constraints, we find that pessimistic risk profiles can be used to reduce cost while improving total reward accumulation. With cost constraints, they are seen to provide higher positive rewards than risk-neutral approaches at the prescribed allowable cost.

Introduction

While deep reinforcement learning (DRL) has been used to master an impressive array of simulated tasks in controlled settings, it has not yet been widely adopted for high-stakes, real-world applications. One reason for this gap is its lack of safety assurances. Endowing artificial agents with a distributional perspective, potentially used in conjunction with cost constraints, should make their decision-making more robust. This could in turn lead to increased trust from humans and increased real-world adoption.

In reinforcement learning (RL), *risk* arises due to uncertainty around the possible outcomes of an agent's actions. It is a result of randomness in the operating environment, mismatch between training and test conditions, and the stochasticity of the policy. *Risk-sensitive* policies, or those that consider a quantity other than the mean reward over the distribution of possible outcomes, offer the potential for added robustness under uncertain and dynamic conditions. There is an evolving landscape of algorithmic paradigms for handling risk in RL, including distributional methods (Bellemare, Dabney, and Munos 2017; Dabney et al. 2018a,b; Barth-Maron et al. 2018; Fei et al. 2021) and constraint-based approaches adapted from optimal control (Bhatnagar 2010; Achiam et al. 2017; Chow et al. 2019; Ray, Achiam, and Amodei 2019; Tessler, Mankowitz, and Mannor 2019; Zhong et al. 2020; Zhang, Vuong, and Ross 2020). Within this landscape, learning approaches that optimize distributional measures offer the ability to express design preferences over the full distribution of potential outcomes. Constraint-based approaches allow the level of *average* cost incurred by an agent to be adjusted, typically through the use of dual methods.

In the following, we introduce a novel method for estimating the policy gradient of a broad class of risk-sensitive objectives, applicable in both the unconstrained and constrained settings. The approach allows agents to be trained with different risk profiles, based on full episode outcomes. It can be used to mimic general human decision-making (Tversky and Kahneman 1992), but is found to be most effective when implementing one particular human learning strategy: emphasizing improvement on tasks where one is deficient. We elucidate the mechanisms behind performance gains associated with this strategy and evaluate its effectiveness for several stochastic continuous control problems.

Related Work

The presence of stochasticity in Markov Decision Processes (MDPs) can lead to variability in agent outcomes. Randomness can come from different sources- for instance the initial state of the environment, noise in transition dynamics, and sampling from a stochastic policy. Distributional RL methods allow for consideration of outcome variability in formulating policies, and have primarily been explored from a value-based perspective. For example, Q-value distribu-

tions have been explicitly modeled through categorical techniques (Bellemare, Dabney, and Munos 2017) and quantile regression (Dabney et al. 2018a), leading to improved value predictions and overall performance. Recent works utilize distribution modeling in the actor-critic setting to enable application to continuous action spaces, again demonstrating improved performance over baseline approaches (Barth-Maron et al. 2018; Ma et al. 2020; Zhang et al. 2021; Duan et al. 2021). In value-based approaches, risk-sensitivity criteria are applied at run time as a nonlinear warping of the estimated Q-value distribution.

Policy optimization with a distributional objective offers additional promise for risk-sensitive RL. Some existing methods are limited to a specific class of learning objective, such as the set of concave risk measures that permit a globally-optimal solution (Zhong et al. 2020; Tamar et al. 2015). Others allow a broader class of measures but are more restrictive in the class of policies that can be represented (Prashanth et al. 2016; Prashanth and Fu 2018, 2022; Jaimungal et al. 2022). Our contribution is a risk-sensitive policy gradient approach that offers both significant flexibility in the choice of learning objective and the ability to learn policies parameterized by a deep neural network. The unconstrained version of the algorithm resembles Proximal Policy Optimization (PPO; Schulman et al. (2017b)) and is similarly widely applicable.

Various measures have been considered in the context of risk-sensitive RL, including exponential utility (Pratt 1964), percentile performance criteria (Wu and Lin 1999), value-at-risk (Leavens 1945), conditional value-at-risk (Rockafellar and Uryasev 2000), and prospect theory (Kahneman and Tversky 1979). In this work, we consider a class of risk-sensitivity measures motivated by Cumulative Prospect Theory (CPT) (Tversky and Kahneman 1992). CPT uniquely models two key aspects of human decision-making: (1) a utility function u, computed relative to a reference point and inducing more risk-averse behavior in the presence of gains than losses as well as (2) a weight function w that prioritizes outlying events. Specific forms of u and w are given in (Tversky and Kahneman 1992); while we evaluate these specific choices we also consider the much broader class of measures possible with different choices. In this work, we typically take u to be the reward provided by the environment and evaluate the effect of adjusting w.

Constrained reinforcement learning addresses safety concerns (García and Fernández 2015) explicitly via methods including Lagrangian constraints (Bhatnagar 2010; Ray, Achiam, and Amodei 2019; Tessler, Mankowitz, and Mannor 2019; Zhang, Vuong, and Ross 2020; Prashanth and Fu 2018; Paternain et al. 2019) and constraint coefficients (Achiam et al. 2017). In this work, we pair our risk-sensitive policy gradient estimate with Reward Constrained Policy Optimization (RCPO; Tessler, Mankowitz, and Mannor (2019)) in order to achieve higher accumulation of non-cost rewards than possible with a risk-neutral objective.

Risk-Sensitive Policy Optimization

In this section we formalize the class of distributional objectives to be considered, derive a sampling-based approximation of its policy gradient, enact variance reduction and regularization on this estimate, and use the result to produce practical learning algorithms for both the unconstrained and constrained settings.

Preliminaries: Problem and Notation

Standard deep reinforcement learning seeks to maximize the expected reward of an agent acting in an MDP. That is, it maximizes the objective

$$J(\theta) = E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]. \tag{1}$$

Here $p_\theta(\tau)$ is the distribution over trajectories $\tau \equiv \mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T$ induced by a policy parameterized by θ; $\mathbf{s}_t$, $\mathbf{a}_t$, and $r(\mathbf{s}_t, \mathbf{a}_t)$ denote the state, action, and reward at time t, respectively. To allow a mapping from reward to utility and outcomes to be weighed based on their relative quality, we instead consider the risk-sensitive objective

$$J_{rs}(\theta) = \int_{-\infty}^{+\infty} u(r(\tau)) \frac{d}{dr(\tau)} \Big(w(P_\theta(r(\tau))) \Big) dr(\tau), \tag{2}$$

where $u(r(\tau))$ is the utility associated with full-trajectory reward $r(\tau) \equiv \sum_t r(\mathbf{s}_t, \mathbf{a}_t)$ and w is a piecewise differentiable weighting function of the CDF of $r(\tau)$; $P_\theta(r(\tau)) = \int_{-\infty}^{r(\tau)} p_\theta(r') dr'$. We assume that the temporal allocation of utility– whether mapped from reward throughout an episode or provided only at the end– is additionally specified.

Equation 2 is inspired by CPT (Tversky and Kahneman 1992), which includes a pair of integrals of this form. It was chosen for its generality; by using different utility functions u and/or weight functions w one may represent all of the risk measures mentioned above, all of the risk measures evaluated by Dabney et al. (2018a), and many more. The form (2) reduces to (1) when u and w are both the identity mapping. It accommodates "cutoff" risk measures (including CVaR) through the use of piecewise weight functions. While designed for the episodic setting, the objective (2) may be considered for infinite horizons through the use of appropriately long windows.

Risk-Sensitive Policy Gradient

To optimize the objective (2), we first derive an approximation to its gradient with respect to the policy parameters θ. Working toward a representation that can be sampled, we assert the independence of the reward on θ and use the chain rule to write

$$\nabla_\theta J_{rs}(\theta) = \int_{-\infty}^{\infty} u(r(\tau)) \frac{d}{dr(\tau)} \Big(w'(P_\theta(r(\tau))) \nabla_\theta P_\theta(r(\tau)) \Big) dr(\tau), \tag{3}$$

where w' is the derivative of w with respect to $P_\theta(r(\tau))$. The gradient of the CDF may be written as:

$$\begin{aligned}\nabla_\theta P_\theta(r(\tau)) &= \nabla_\theta \int_{-\infty}^{r(\tau)} p_\theta(r')dr' \\ &= \nabla_\theta \int_{\tau'} H(r(\tau) - r(\tau'))p_\theta(\tau')d\tau' \\ &= \int_{\tau'} H(r(\tau) - r(\tau'))\nabla_\theta p_\theta(\tau')d\tau' \\ &= \int_{\tau'} H(r(\tau) - r(\tau'))p_\theta(\tau')\nabla_\theta \log p_\theta(\tau')d\tau'.\end{aligned} \tag{4}$$

Here we have used the integral representation of $P_\theta(r(\tau))$, the Heaviside step function H to select all trajectories with total reward $\leq r(\tau)$, and the independence of reward on θ. In the following, we also use the complementary expression

$$\begin{aligned}\nabla_\theta P_\theta(r(\tau)) &= \nabla_\theta \left(1 - \int_{r(\tau)}^{\infty} p_\theta(r')dr'\right) \\ &= -\int_{\tau'} H(r(\tau') - r(\tau))p_\theta(\tau')\nabla_\theta \log p_\theta(\tau')d\tau'.\end{aligned} \tag{5}$$

Either form, or a combination of the two, may be substituted into (3) and the result sampled over N trajectories by first ordering trajectories $i = 1 \dots N$ by increasing reward $r(\tau)$. Implicitly, this assumes that the collected full-episode rewards are representative of the true distribution. Then

$$\begin{aligned}\nabla_\theta J_{rs}(\theta) \approx \sum_{i=1}^{N} u(r(\tau_i)) \Bigg(& w'\left(\frac{i}{N}\right)\nabla_\theta P_\theta(r(\tau_i)) \\ & - w'\left(\frac{i-1}{N}\right)\nabla_\theta P_\theta(r(\tau_{i-1}))\Bigg),\end{aligned} \tag{6}$$

where we have discretized $d/dr(\tau)$ and the term $w'(0)\nabla_\theta P_\theta(r(\tau_0)) \equiv 0$. Such ordering produces an asymptotically consistent estimate of the CPT value (Prashanth et al. 2016). $\nabla_\theta P_\theta(r(\tau_i))$ may be sampled in one of two ways, based on either (4) or (5):

$$\begin{aligned}\nabla_\theta P_\theta(r(\tau_i)) &\approx \frac{1}{N}\sum_{j=1}^{i}\sum_{t=1}^{T_j} \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \\ &\approx -\frac{1}{N}\sum_{j=i+1}^{N}\sum_{t=1}^{T_j} \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}).\end{aligned} \tag{7}$$

The expression (6) may be used to train a policy that optimizes the distributional objective (2) in a manner similar to REINFORCE (Williams 1992).

Variance Reduction and Regularization

Reducing the variance of sample-based gradient estimates enables faster learning. Here we take several steps to reduce the variance of (6), similar to what has been done with the policy gradient estimate of REINFORCE (Williams 1992). First, note that cross-trajectory terms of the form $f(\tau_i, \mathbf{a}_{j,t}, \mathbf{s}_{j,t}) \equiv u(r(\tau_i))\nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t})$, while nonzero, do not contribute to the gradient estimate in expectation when $i \neq j$. A proof of this assertion (relevant to our approach but not REINFORCE), is given in Appendix A.1 of Markowitz et al. (2022). Using (4) for the first term of (6) and (5) for the second allows us to write

$$\begin{aligned}\nabla_\theta J_{rs}(\theta) \approx \sum_{i=1}^{N} u(r(\tau_i)) * \\ \Bigg(w'\left(\frac{i}{N}\right)\frac{1}{N}\sum_{j=1}^{i}\sum_{t=1}^{T_j}\nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \\ + w'\left(\frac{i-1}{N}\right)\frac{1}{N}\sum_{j=i}^{N}\sum_{t=1}^{T_j}\nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t})\Bigg).\end{aligned} \tag{8}$$

Removing cross-trajectory terms gives

$$\begin{aligned}\nabla_\theta J_{rs}(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} u(r(\tau_i)) * \\ \left(w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right)\right)\sum_{t=1}^{T_i}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}).\end{aligned} \tag{9}$$

Note that the weight coefficients $(w'(\frac{i}{N}) + w'(\frac{i-1}{N}))$ should be normalized over each batch. The expression (9) is equal to (6) in expectation, but with reduced variance (justification in Appendix A.1 of Markowitz et al. (2022)). It has a clear intuition – trajectories are assigned utilities based on their rewards and their contributions to the policy gradient are scaled by the derivative of the weight function, just as they are in CPT (Tversky and Kahneman 1992).

Standard variance reduction techniques may be applied to this simplified form. Without further assumption or introduction of additional bias, a static baseline b may be employed:

$$\begin{aligned}\nabla_\theta J_{rs}(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(u(r(\tau_i)) - b\right) * \\ \left(w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right)\right)\sum_{t=1}^{T_i}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}).\end{aligned} \tag{10}$$

Justification for this assertion is given in Appendix A.2 of Markowitz et al. (2022). Learning may be further expedited by considering utilities on a per-step basis. Given our assumptions that utility is a function of only reward and that its temporal allocation is given with the objective, we may further reduce the variance of (9) through the incorporation of utility-to-go and a state-dependent baseline $V_\phi(\mathbf{s}_{i,t})$:

$$\begin{aligned}\nabla_\theta J_{rs}(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left[w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right)\right] * \\ \sum_{t=1}^{T_i}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})\left[\sum_{t'=t}^{T_i} u(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) - V_\phi(\mathbf{s}_{i,t})\right]\end{aligned} \tag{11}$$

Here $u(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'})$ is the per-step utility. The value function $V_\phi(\mathbf{s}_{i,t})$ is parameterized by ϕ and may be trained via regression to minimize

$$\mathcal{L}(\phi) = \sum_{i,t} \left(V_\phi(\mathbf{s}_{i,t}) - \sum_{t'=t}^{T_i} u(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) \right)^2. \quad (12)$$

A standard argument, similar to the approach taken in (Achiam 2018), can be used to show that the incorporation of utility-to-go does not change the expected value of (9). The use of a state-dependent baseline also does not introduce additional bias (see Appendix A.2 of Markowitz et al. (2022)).

Finally, discount factors, bootstrapping, and trust regions may be used to provide additional variance reduction and regularization (see Appendix A.2 of Markowitz et al. (2022)). These measures may introduce additional bias to the policy gradient estimate, but typically lead to more sample-efficient learning. In our experiments, we evaluate the use of generalized advantage estimation (GAE; (Schulman et al. 2016)) based on utility-to-go as well as clipping-based regularization similar to Proximal Policy Optimization (Schulman et al. 2017b). Incorporating these in our policy gradient estimate yields

$$\nabla_\theta J_{rs}(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right) \right) * \sum_{t=1}^{T_i} \nabla_\theta L_{\text{clip}}\Big(\log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}), A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \Big), \quad (13)$$

where $A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$ is the standard GAE except with per-step utilities in place of rewards. Trust regions are implemented similarly to PPO, pessimistically clipping policy updates to be within a multiplicative factor of $1 \pm \epsilon$ of the existing policy:

$$L_{\text{clip}} = \min \Bigg(\log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}), \log \left(\text{clip}\left(\frac{\pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})}{\pi_{\theta_{\text{old}}}(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})}, 1 \pm \epsilon \right) \pi_{\theta_{\text{old}}}(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \right) A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \Bigg). \quad (14)$$

The form of this clipping differs slightly from that of PPO, due to the difference between our policy gradient and the gradient of the objective used by PPO. In practice, we found our form to consistently perform better (compare blue and green traces in Figures 3 and 4 and see Markowitz et al. (2022) for further discussion). As in PPO, our clipping can be used to perform multiple policy updates with the same batch of data, significantly improving sample efficiency. When following this route, we apply early stopping based on the Kullback-Leibler divergence (D_{KL}) between old and new policies, as in (Ray, Achiam, and Amodei 2019).

Finally we note that, in the case of policy distributions with infinite support, the "clipped action policy gradient" correction of (Fujita and Maeda 2018) should be used to properly handle finite control bounds. This was done for all methods (baselines included) in our experiments.

Application in Constrained Settings

The policy gradient estimate derived above may also be used to maximize a risk-sensitive objective subject to a constraint. Constrained Markov Decision Processes (CMDPs) have positive rewards $r(\mathbf{s}, \mathbf{a})$ and costs $c(\mathbf{s}, \mathbf{a})$ defined for each time step, as well as an overall constraint $C(\tau) = F(c(\mathbf{s}_1, \mathbf{a}_1), \ldots, c(\mathbf{s}_T, \mathbf{a}_T))$ defined over the whole trajectory. The associated learning problem is to find

$$\max_\theta J_R(\theta) \text{ s.t. } J_C(\theta) \leq d, \quad (15)$$

where $J_R(\theta)$ is the objective based on positive reward, $J_C(\theta) = E_{\tau \sim p_\theta(\tau)} C(\tau)$, and d is a fixed threshold. Reward Constrained Policy Optimization (RCPO; (Tessler, Mankowitz, and Mannor 2019)) is a recent method for learning CMDP policies. RCPO learns to scale the weight of cost terms relative to rewards by solving a dual problem, treating the scaling factor as a Lagrange Multiplier.

A constrained, risk-sensitive learner may be formulated by using our risk-sensitive policy gradient estimate in a formulation resembling RCPO. Our motivation for doing this is twofold. First, it allows for an acceptable cost limit to be set ahead of time, removing the need to manually tune the relative weights of positive and negative reward terms. Second, it leverages the observed tendency of our "pessimistic" agents to accumulate lower costs at similar or higher positive reward levels than standard approaches.

Learning Algorithm

The above policy gradient estimate may be used to maximize distributional objectives of the form (2), with or without constraints. The constrained method, Constrained, Risk-Sensitive Proximal (CRiSP) policy optimization, is given in Algorithm 1. Note that the Adam optimizer (Kingma and Ba 2015) was used for all parameter sets. To perform unconstrained learning, one would simply fix λ and combine the positive and negative reward terms into a single function $r + c$ (thereby allowing a single value function). The unconstrained method is given explicitly in Appendix A.3 of Markowitz et al. (2022).

Algorithm 1 differs from conventional methods in the requirement to collect full episodes of data in each batch. This is unnecessary if outcomes can be defined over partial episodes, an assumption that is often viable (for instance with the Atari suite (Bellemare et al. 2013)) and matches human decision-making.

Experiments

We used the OpenAI Safety Gym (Ray, Achiam, and Amodei 2019) to evaluate our approach. Safety Gym is a configurable suite of continuous, multidimensional control tasks wherein different types of robots must navigate through obstacles with different dynamics to perform different tasks. By including both positive and negative reward terms, it allows evaluation of how agents handle risk and constraints. Safety Gym is also highly stochastic: the locations of the goals and obstacles are randomized, leading to outcome variability and requiring a generalized strategy.

Algorithm 1: CRiSP Policy Optimization

Require: Policy: initial parameters θ_0, learning rate α_θ, updates per batch M_θ
Require: Value functions: initial parameters for utility, cost value functions $\phi_{u,0}$, $\phi_{c,0}$, steps per update M_{ϕ_u}, M_{ϕ_c}
Require: Penalty: initial value $\lambda_0 \geq 0$, learning rate α_λ
Require: Stopping threshold $D_{\text{KL, stop}}$, discount factor γ

for $k = 0, 1, 2, \ldots$ **do**
  Collect set of episodes $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi(\theta_k)$ in the environment
  Update penalty $\lambda_{k+1} = \lambda_k + \alpha_\lambda(J_C(\theta) - d)$, using cost constraint $J_C(\theta)$ and limit d (1 step)
  Compute discounted utilities-to-go:
  $$\hat{u}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) = \sum_{t'=t}^{T_i} \gamma^{t'-t} u(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'})$$
  Fit utility value function (M_{ϕ_u} steps):
  $$\phi_{u,k+1} = \arg\min_{\phi_u} \frac{1}{\sum_i T_i} \sum_{i,t} \left(V_{\phi_u}(\mathbf{s}_{i,t}) - \hat{u}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)^2$$
  Compute discounted cost-to-go:
  $$\hat{c}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) = \sum_{t'=t}^{T_i} \gamma^{t'-t} c(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'})$$
  Fit cost value function (M_{ϕ_c} steps):
  $$\phi_{c,k+1} = \arg\min_{\phi_c} \frac{1}{\sum_i T_i} \sum_{i,t} \left(V_{\phi_c}(\mathbf{s}_{i,t}) - \hat{c}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)^2$$
  Update effective utility in batch:
  $$u(\mathbf{s}_t, \mathbf{a}_t) \leftarrow u(\mathbf{s}_t, \mathbf{a}_t) - \lambda c(\mathbf{s}_t, \mathbf{a}_t)$$
  Update utility advantage estimates $A^\pi_u(\mathbf{s}, \mathbf{a})$ using $V_\phi(\mathbf{s}) = V_{\phi_u}(\mathbf{s}) - \lambda V_{\phi_c}(\mathbf{s})$
  Compute weight coefficients based on ordered full-episode rewards
  Update policy using clipped-action policy gradient correction over M_θ steps with KL-based early stopping (threshold $D_{\text{KL, stop}}$):
  $$\theta_{k+1} = \arg\max_\theta \left(\frac{1}{N} \sum_{i=1}^{N} \left(w'\left(\tfrac{i}{N}\right) + w'\left(\tfrac{i-1}{N}\right) \right) * \sum_{t=1}^{T_i} L_{\text{clip}}(\log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}), A^\pi_u(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})) \right)$$
end for

Safety Gym logs adverse events but does not include them in the reward function. For unconstrained experiments, we assigned each logged adverse event a fixed, negative reward. The coefficient for cost events was learned in constrained experiments. To highlight performance variability, we focused on the most obstacle-rich (level 2) publicly available environments. Avoiding the longer compute time of the "Doggo" robot, we evaluated the "Point" and "Car" robots on each task ("Goal", "Button", and "Push"). Additional details are available in Appendix A.4 of Markowitz et al. (2022).

In all experiments, we evaluated five random seeds and matched the hyperparameters used in the baselines accompanying Safety Gym as closely as possible. The neural networks used to model both policy and value were multilayer perceptrons (MLPs), with two hidden layers of 256 units each and $\tanh$ activations. The policy networks output the mean values of a multivariate gaussian with diagonal covariance. Control variances were optimized but independent of state. All variance reduction and regularization measures were used throughout (see Appendix A.5 of Markowitz et al. (2022) for experimental justification).

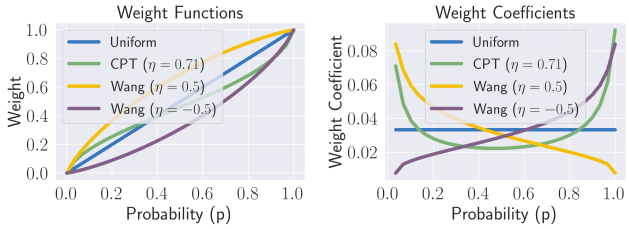

Figure 1: Example weight functions and their resulting coefficients in the policy gradient estimate (9).

One drawback of our approach is the additional computational expense of its sorting of full-episode rewards (30 per batch here). In practice we found this to lead to only minor slowdowns. On average, our method trained 13% slower than PPO in unconstrained trials and 16% slower in constrained trials.

Differing Objectives

Agent performance was first explored under four different distributional objectives. In addition to expected reward and CPT (configured to match the original form of Tversky and Kahneman (1992) and as given in Appendix A.5 of Markowitz et al. (2022)), we optimized for pessimistic ($\eta = 0.5$) and optimistic ($\eta = -0.5$) versions of the distortion risk measure proposed in (Wang 2000). This measure is defined as $w(p) = \Phi(\Phi^{-1}(p) + \eta)$, where Φ and Φ^{-1} are the standard normal cumulative distribution function and its inverse. While this form is convenient, the "Pow" metric in (Dabney et al. 2018a) or any other set of similarly shaped w curves should produce a similar effect. Note that only the CPT objective used a non-identity mapping from reward to utility. The four weight functions and their corresponding coefficients in (9) are shown in Figure 1. The effects of varying η on weight functions and coefficients are displayed more fully in Appendix A.9 of Markowitz et al. (2022).

What Should We Expect?

Before presenting results, it is instructive to inspect the form (13) to set expectations on the impact of different weightings. The standard policy gradient of (Williams 1992) can be thought of as maximum likelihood estimation weighted by advantage over observed trajectories. Over time, it shifts probability mass toward states with positive advantage and away from states with negative advantage. Weighing gradient contributions differently based on episode outcome adjusts this migration. In particular, emphasizing experiences from low-reward episodes (pessimistic weighting) should on average lead to stronger contributions from low-advantage terms. This prioritizes pushing probability mass away from problematic parts of state space, increasing policy entropy. Conversely, emphasizing experiences from high-reward episodes (optimistic weighting) should prioritize pushing probability mass toward advantageous parts of state space, more rapidly decreasing policy entropy. One

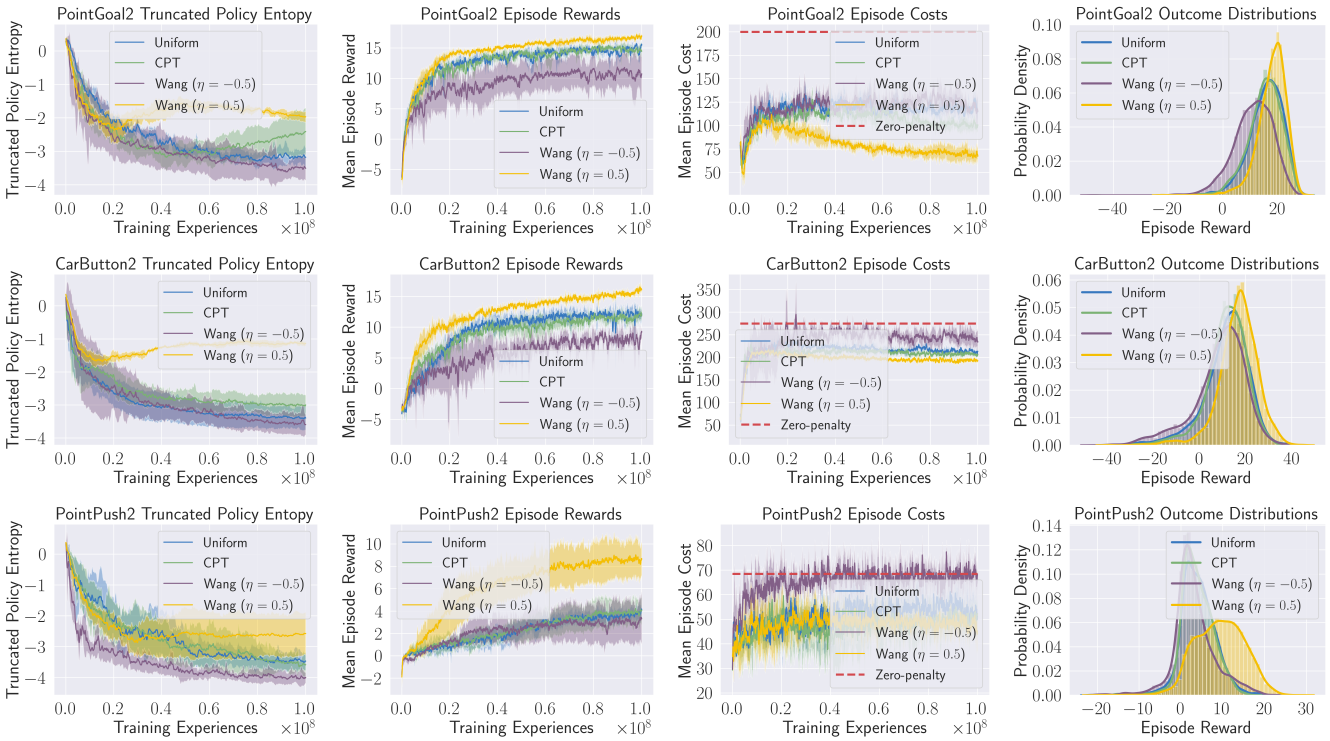

Figure 2: Impact of different distributional objectives in three environments. Each row represents an environment. First column: pessimistic weightings ($\eta > 0$; yellow) lead to higher policy entropy, aiding exploration. Second and third columns: pessimistic weightings lead to higher overall reward (positive reward minus cost) with lower cost contributions than other weightings. Fourth column: testing distributions across seeds (sampling turned off).

.

should therefore expect pessimistic weightings to more thoroughly explore the state space, potentially avoiding premature convergence to suboptimal policies.

As a simple quantitative example, consider the contribution to the gradient of the variance parameters of state-independent Gaussian policies for trajectory i, to control dimension j, at time step t:

$$\nabla_{\theta_{\sigma_j}} \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) A(\mathbf{s}_{i,t,}, \mathbf{a}_{i,t}) = \frac{1}{\sigma_j}\left(\frac{(\mathbf{a}_{i,t,j} - \mu_{i,t,j})^2}{\sigma_j^2} - 1\right) A(\mathbf{s}_{i,t,}, \mathbf{a}_{i,t}). \quad (16)$$

Here we use the chain rule and the fact that σ is a single parameter in each dimension. Note that the first multiplicative term in the σ derivative is always positive and the second is negative about 68% of the time (and more negative when sampling near μ). As learning occurs, sampling close to μ should correlate with higher advantages. These trends combine to produce the decrease in σ observed in standard learning. When pessimistic weightings emphasize terms with negative advantage, however, the decrease in σ may be slowed or even reversed.

Impact of Pessimistic Weightings

Figure 2 shows the impact of different weightings when training in three environments. As expected, we observe that policy variance is higher when using pessimistic weightings (Markowitz et al. 2022). This typically leads to higher truncated (considering control bounds) entropy, depending on how often learned actions are near the boundaries. Higher levels of total rewards and lower cost levels were achieved with pessimistic weightings, both in training and testing (i.e., with sampling turned off). Note that in this and subsequent plots, the "zero penalty" lines reflect the cost levels that standard PPO and TRPO agents reach when unaware of cost (Ray, Achiam, and Amodei 2019). Finally, as shown in Markowitz et al. (2022), pessimistic weightings produced higher levels of all metrics in all environments in testing.

Comparison with Other Unconstrained Baselines

We pursued comparisons of our risk-sensitive approach, using the pessimistic objective from (Wang 2000), to state-of-the-art on-policy methods. In addition to PPO, we compared performance with Trust Region Policy Optimization (TRPO; Schulman et al. (2017a)). Both were configured as in (Ray, Achiam, and Amodei 2019). As shown in Figure 3 and Appendix A.7 of Markowitz et al. (2022), a single pessimistic objective ($\eta = 0.5$) could be used to provide both higher total reward (sum of positive and negative terms) and lower cost than PPO and TRPO in five of the six environments. A less aggressive weighting ($\eta = 0.25$) provided these gains in the last environment.

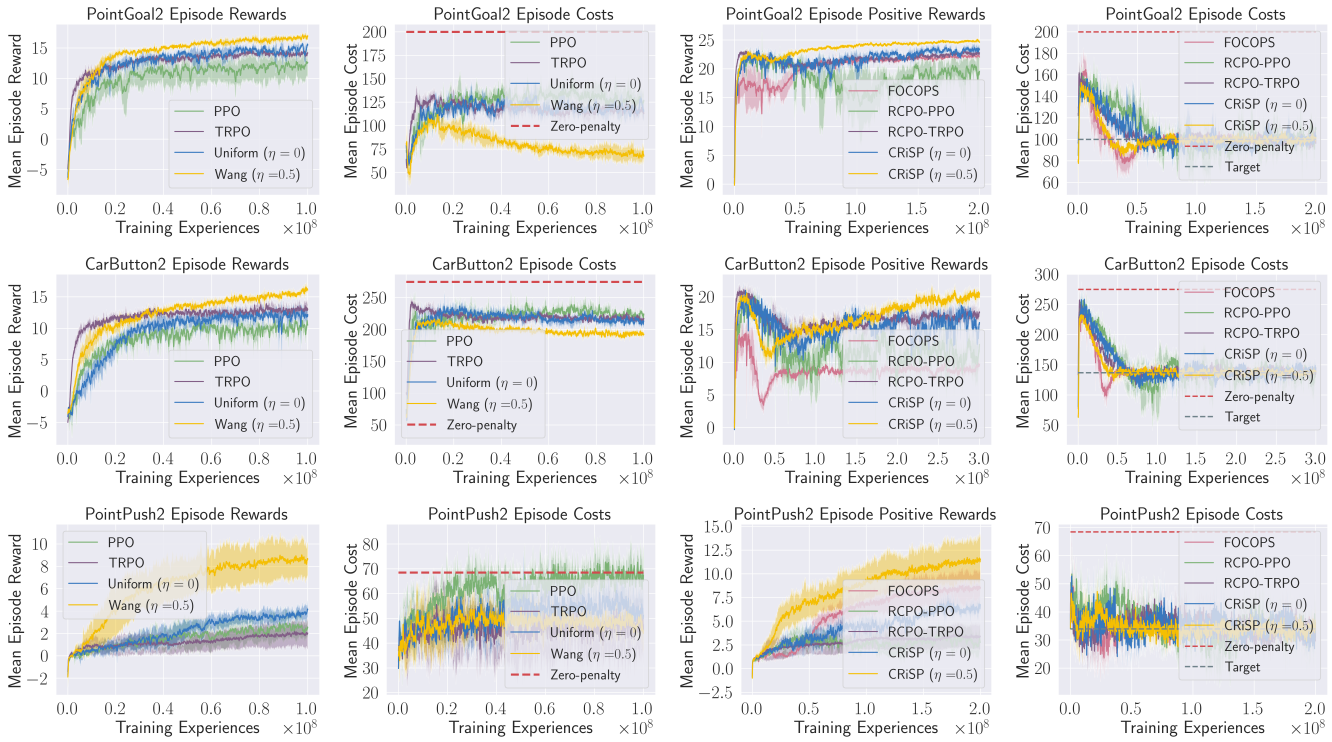

Figure 3: Comparison of our pessimistic agents with $\eta = 0.5$ (yellow) to other on-policy methods. Columns 1-2: In unconstrained learning, pessimistic agents tend toward higher total reward (including penalty) and lower cost than others. Columns 3-4: In the constrained setting, pessimistic agents accumulate more positive reward than others at the target cost level.

Comparisons with Constrained Baselines

We additionally compared the performance of the constrained version of our approach (CRiSP) to RCPO using PPO and TRPO updates as well as First-Order Constrained Optimization in Policy Space (FOCOPS; Zhang, Vuong, and Ross (2020)). These baselines were found to be significantly stronger than the constrained methods explored in (Ray, Achiam, and Amodei 2019); see (Tessler, Mankowitz, and Mannor 2019) for a likely explanation. For all tasks, we chose the cost target to be half of what a trained, unconstrained agent unaware of penalties would accumulate.

Results are given in Figure 3 and Appendix A.8 of Markowitz et al. (2022). Our pessimistic agents are seen to achieve higher positive rewards than all other methods, at the same cost levels, in all environments tested. Only FOCOPS consistently allows lower learned penalty coefficients and higher policy entropies than CRiSP; however FOCOPS does not match the positive reward accumulation of our method. In Appendix A.9 of Markowitz et al. (2022), we show that η may be increased to hasten convergence to the target cost (though this eventually reduces reward accumulation).

Discussion

As shown above, pessimistic agents consistently achieve superior performance in our formulation. One contributing factor is their enhanced exploration, which reduces the chance of premature convergence to a suboptimal policy. However this cannot be the only factor, as evidenced by the consistently higher entropy and lower performance of the FOCOPS algorithm (see Appendix A.8 of Markowitz et al. (2022)). The fact that pessimistic agents emphasize poor outcomes likely plays a significant role, as it allows behavior to continually be adjusted most where it is most necessary. Once a problematic part of the state space is addressed, a different region takes its place. Conversely, optimistic weightings emphasize the best outcomes in the distribution. Already strong outcomes are given increased attention, making them likely to stay on top. Agents trained optimistically thus become myopic, obsessing over a fraction of the state space while neglecting the rest of it.

While $\eta \geq 0$ produced gains in all environments tested, it does represent an additional hyperparameter. Our results suggest that a reasonable strategy for choosing it is to start at $\eta = 0.5$ and proceed downward toward $\eta = 0$ if needed.

Future directions based on these findings include an off-policy formulation (to improve sample efficiency) and additional experiments to clarify the impact of pessimistic weightings in discrete action spaces.

Conclusions

We formulated unconstrained and constrained learning based on a risk-sensitive policy gradient estimate. Objectives that emphasize improvement where performance is poor produced performance gains in all environments tested.

Acknowledgements

This work was funded by the Johns Hopkins Institute for Assured Autonomy. We would like to thank the AAAI reviewers for their constructive feedback.

References

Achiam, J. 2018. OpenAI Spinning Up: Proof for Don't Let the Past Distract You. http://spinningup.openai.com/en/latest/spinningup/extra_pg_proof1.html. Accessed: 2022-01-27.

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 22–31.

Barth-Maron, G.; Hoffman, M. W.; Budden, D.; Dabney, W.; Horgan, D.; TB, D.; Muldal, A.; Heess, N.; and Lillicrap, T. 2018. Distributional Policy Gradients. In *International Conference on Learning Representations*.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A Distributional Perspective on Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, 449–458.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47: 253–279.

Bhatnagar, S. 2010. An actor–critic algorithm with function approximation for discounted cost constrained Markov decision processes. *Systems & Control Letters*, 59(12): 760–766.

Chow, Y.; Nachum, O.; Faust, A.; Ghavamzadeh, M.; and Duéñez-Guzmán, E. A. 2019. Lyapunov-based Safe Policy Optimization for Continuous Control. *CoRR*, abs/1901.10031.

Dabney, W.; Ostrovski, G.; Silver, D.; and Munos, R. 2018a. Implicit Quantile Networks for Distributional Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, 1096–1105.

Dabney, W.; Rowland, M.; Bellemare, M. G.; and Munos, R. 2018b. Distributional Reinforcement Learning With Quantile Regression. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2892–2901.

Duan, J.; Guan, Y.; Li, S. E.; Ren, Y.; Sun, Q.; and Cheng, B. 2021. Distributional Soft Actor-Critic: Off-Policy Reinforcement Learning for Addressing Value Estimation Errors. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Fei, Y.; Yang, Z.; Chen, Y.; and Wang, Z. 2021. Exponential Bellman Equation and Improved Regret Bounds for Risk-Sensitive Reinforcement Learning. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 20436–20446. Curran Associates, Inc.

Fujita, Y.; and Maeda, S.-i. 2018. Clipped Action Policy Gradient. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1597–1606. PMLR.

García, J.; and Fernández, F. 2015. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, 16(42): 1437–1480.

Jaimungal, S.; Pesenti, S. M.; Wang, Y. S.; and Tatsat, H. 2022. Robust Risk-Aware Reinforcement Learning. *SIAM Journal on Financial Mathematics*, 13(1): 213–226.

Kahneman, D.; and Tversky, A. 1979. Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, 47(2): 263–291.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.

Leavens, D. H. 1945. Diversification of Investments. *Trusts and Estates*, 80: 469–473.

Ma, X.; Xia, L.; Zhou, Z.; Yang, J.; and Zhao, Q. 2020. DSAC: Distributional Soft Actor Critic for Risk-Sensitive Reinforcement Learning. ArXiv:2004.14547.

Markowitz, J.; Gardner, R. W.; Llorens, A.; Arora, R.; and Wang, I.-J. 2022. A Risk-Sensitive Approach to Policy Optimization. ArXiv:2208.09106.

Paternain, S.; Chamon, L.; Calvo-Fullana, M.; and Ribeiro, A. 2019. Constrained Reinforcement Learning Has Zero Duality Gap. In *Neural Information Processing Systems (NeurIPS)*.

Prashanth, L.; and Fu, M. 2022. *Risk-sensitive Reinforcement Learning Via Policy Gradient Search*. Foundations and trends in machine learning. Now Publishers. ISBN 9781638280279.

Prashanth, L.; and Fu, M. C. 2018. Risk-Sensitive Reinforcement Learning: A Constrained Optimization Viewpoint. *CoRR*, abs/1810.09126.

Prashanth, L.; Jie, C.; Fu, M. C.; Marcus, S. I.; and Szepesvári, C. 2016. Cumulative Prospect Theory Meets Reinforcement Learning: Prediction and Control. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, 1406–1415.

Pratt, J. W. 1964. Risk Aversion in the Small and in the Large. *Econometrica*, 32: 122–136.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking Safe Exploration in Deep Reinforcement Learning. https://cdn.openai.com/safexp-short.pdf. Accessed: 2022-01-27.

Rockafellar, R. T.; and Uryasev, S. 2000. Optimization of Conditional Value-at-Risk. *Journal of Risk*, 2: 21–41.

Schulman, J.; Levine, S.; Moritz, P.; Jordan, M. I.; and Abbeel, P. 2017a. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017b. Proximal Policy Optimization Algorithms. ArXiv:1707.06347.

Tamar, A.; Chow, Y.; Ghavamzadeh, M.; and Mannor, S. 2015. Policy Gradient for Coherent Risk Measures. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, 1468–1476.

Tessler, C.; Mankowitz, D. J.; and Mannor, S. 2019. Reward Constrained Policy Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tversky, A.; and Kahneman, D. 1992. Advances in Prospect Theory: Cumulative Representation of Uncertainty. *Journal of Risk and Uncertainty*, 5(4): 297–323.

Wang, S. S. 2000. A Class of Distortion Operators for Pricing Financial and Insurance Risks. *The Journal of Risk and Insurance*, 67(1): 15.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement learning. *Machine Learning*, 8(3): 229–256.

Wu, C.; and Lin, Y. 1999. Minimizing Risk Models in Markov Decision Processes with Policies Depending on Target Values. *Journal of Mathematical Analysis and Applications*, 231: 47–67.

Zhang, J.; Bedi, A. S.; Wang, M.; and Koppel, A. 2021. Cautious Reinforcement Learning via Distributional Risk in the Dual Domain. *IEEE Journal on Selected Areas in Information Theory*, 2(2): 611–626.

Zhang, Y.; Vuong, Q.; and Ross, K. 2020. First Order Constrained Optimization in Policy Space. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 15338–15349. Curran Associates, Inc.

Zhong, H.; Fang, E. X.; Yang, Z.; and Wang, Z. 2020. Risk-Sensitive Deep RL: Variance-Constrained Actor-Critic Provably Finds Globally Optimal Policy. ArXiv:2012.14098.