Uncertainty-Aware Policy Optimization: A Robust, Adaptive Trust Region Approach

James Queeney, Ioannis Ch. Paschalidis, Christos G. Cassandras
Division of Systems Engineering
Boston University
Boston, MA 02215
{jqueeney, yannisp, cgc}@bu.edu

Abstract

In order for reinforcement learning techniques to be useful in real-world decision making processes, they must be able to produce robust performance from limited data. Deep policy optimization methods have achieved impressive results on complex tasks, but their real-world adoption remains limited because they often require significant amounts of data to succeed. When combined with small sample sizes, these methods can result in unstable learning due to their reliance on high-dimensional sample-based estimates. In this work, we develop techniques to control the uncertainty introduced by these estimates. We leverage these techniques to propose a deep policy optimization approach designed to produce stable performance even when data is scarce. The resulting algorithm, Uncertainty-Aware Trust Region Policy Optimization, generates robust policy updates that adapt to the level of uncertainty present throughout the learning process.

1 Introduction

By combining policy optimization techniques with rich function approximators such as deep neural networks, the field of reinforcement learning has achieved significant success on a variety of high-dimensional continuous control tasks (Duan et al. 2016). Despite these promising results, there are several barriers preventing the widespread adoption of deep reinforcement learning techniques for real-world decision making. Most notably, policy optimization algorithms can exhibit instability during training and high sample complexity, which are further exacerbated by the use of neural network function approximators. These are undesirable qualities in important applications such as robotics and healthcare, where data collection may be expensive and poor performance at any point can be both costly and dangerous.

Trust Region Policy Optimization (TRPO) (Schulman et al. 2015) is one of the most popular methods that has been developed to address these issues, utilizing a trust region in policy space to generate stable but efficient updates. In order to perform well in practice, however, TRPO often requires a large number of samples to be collected prior to each policy update. This is because TRPO, like all policy optimization algorithms, relies on sample-based estimates to approximate expectations. These estimates are known to suffer from high variance, particularly for problems with long time horizons. As a result, the use of sample-based estimates can be a major source of error unless a sufficiently large number of samples are collected.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Motivated by the need to make decisions from limited data in real-world settings, we focus on addressing the instability caused by finite-sample estimation error in policy optimization. By directly accounting for the uncertainty present in sample-based estimates when generating policy updates, we can make efficient, robust use of limited data to produce stable performance throughout the training process. In this work, we develop an algorithm we call *Uncertainty-Aware Trust Region Policy Optimization (UA-TRPO)* that controls the finite-sample estimation error in the two main components of TRPO: (i) the policy gradient and (ii) the trust region metric. Our main contributions are as follows:

1. We construct a finite-sample policy improvement lower bound that is accurate up to first and second order approximations, which we use to motivate an adaptive trust region that directly considers the uncertainty in the policy gradient estimate.
2. We propose a computationally efficient technique to restrict policy updates to a subspace where trust region information is available from the observed trajectories.
3. We demonstrate the robust performance of our approach through experiments on high-dimensional continuous control tasks in OpenAI Gym's MuJoCo environments (Brockman et al. 2016; Todorov, Erez, and Tassa 2012).

We summarize our uncertainty-aware modifications to TRPO in Figure 1.

2 Preliminaries

Notational Conventions. We use bold lowercase letters to denote vectors, bold uppercase letters to denote matrices, script letters to denote sets, and prime to denote transpose. $\mathbb{E}[\cdot]$ represents expectation, and we use hats ($\hat{\cdot}$) to denote sample-based estimates.

Reinforcement Learning Framework. We model the sequential decision making problem as an infinite-horizon, discounted Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \rho_0, \gamma)$. $\mathcal{S}$ is the set of states and $\mathcal{A}$ is the set of actions, both possibly infinite. $p : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$

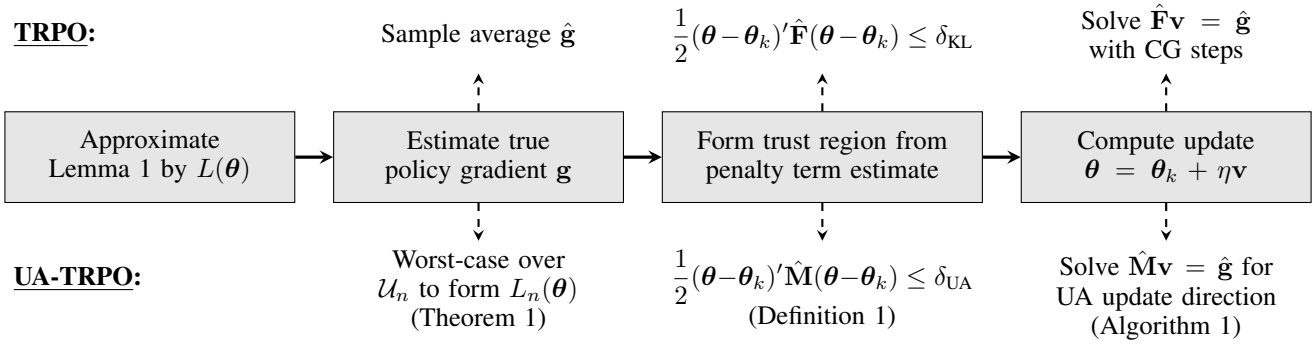

Figure 1: Comparison of TRPO and UA-TRPO. Both algorithms are derived from the lower bound in Lemma 1 and its approximation $L(\boldsymbol{\theta})$. Abbreviations: CG denotes conjugate gradient; UA denotes uncertainty-aware.

is the transition probability function of the MDP where $\Delta_{\mathcal{S}}$ denotes the space of probability distributions over $\mathcal{S}$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\rho_0 \in \Delta_{\mathcal{S}}$ is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor.

We model the agent's decisions as a stationary policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$, where $\pi(a \mid s)$ is the probability of taking action a in state s. In policy optimization, we search over a restricted class of policies $\pi_{\boldsymbol{\theta}} \in \Pi_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$, where a neural network is typically used to represent $\pi_{\boldsymbol{\theta}}$. The standard goal in reinforcement learning is to find a policy parameter $\boldsymbol{\theta}$ that maximizes the expected total discounted reward $J(\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right]$, where $\tau \sim \pi_{\boldsymbol{\theta}}$ represents a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$ sampled according to $s_0 \sim \rho_0(\cdot)$, $a_t \sim \pi_{\boldsymbol{\theta}}(\cdot \mid s_t)$, and $s_{t+1} \sim p(\cdot \mid s_t, a_t)$.

We adopt standard reinforcement learning definitions throughout the paper. We denote the advantage function of $\pi_{\boldsymbol{\theta}}$ as $A_{\boldsymbol{\theta}}(s, a) = Q_{\boldsymbol{\theta}}(s, a) - V_{\boldsymbol{\theta}}(s)$, where $Q_{\boldsymbol{\theta}}(s, a) = \mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right]$ is the state-action value function and $V_{\boldsymbol{\theta}}(s) = \mathbb{E}_{a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[Q_{\boldsymbol{\theta}}(s, a)\right]$ is the state value function. We denote the normalized discounted state visitation distribution induced by $\pi_{\boldsymbol{\theta}}$ as $\rho_{\boldsymbol{\theta}} \in \Delta_{\mathcal{S}}$ where $\rho_{\boldsymbol{\theta}}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s \mid \rho_0, \pi_{\boldsymbol{\theta}}, p)$, and the corresponding normalized discounted state-action visitation distribution as $d_{\boldsymbol{\theta}} \in \Delta_{\mathcal{S}\times\mathcal{A}}$ where $d_{\boldsymbol{\theta}}(s, a) = \rho_{\boldsymbol{\theta}}(s)\pi_{\boldsymbol{\theta}}(a \mid s)$.

Trust Region Policy Optimization. TRPO is motivated by the goal of monotonic improvement used by Kakade and Langford (2002) in Conservative Policy Iteration (CPI). CPI achieves monotonic improvement by maximizing a lower bound on policy improvement that can be constructed using only samples from the current policy. The lower bound developed by Kakade and Langford (2002) applies only to mixture policies, but was later extended to arbitrary policies by Schulman et al. (2015) and further refined by Achiam et al. (2017):

Lemma 1 (Achiam et al. (2017), Corollary 3)**.** *Consider two policies $\pi_{\boldsymbol{\theta}_k}$ and $\pi_{\boldsymbol{\theta}}$. Let $\epsilon_{\boldsymbol{\theta}} = \max_s \left|\mathbb{E}_{a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}\left[A_{\boldsymbol{\theta}_k}(s, a)\right]\right|$, and denote the Kullback-Leibler (KL) divergence between $\pi_{\boldsymbol{\theta}_k}(\cdot \mid s)$ and $\pi_{\boldsymbol{\theta}}(\cdot \mid s)$ by $\text{KL}(\boldsymbol{\theta}_k\|\boldsymbol{\theta})(s)$. The difference between expected total discounted rewards $J(\boldsymbol{\theta}_k)$ and $J(\boldsymbol{\theta})$ can be bounded below by*

$$J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k) \geq \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{(s,a)\sim d_{\boldsymbol{\theta}_k}}\left[\frac{\pi_{\boldsymbol{\theta}}(a \mid s)}{\pi_{\boldsymbol{\theta}_k}(a \mid s)}A_{\boldsymbol{\theta}_k}(s, a)\right] - \frac{\sqrt{2}\gamma\epsilon_{\boldsymbol{\theta}}}{(1-\gamma)^2}\sqrt{\mathop{\mathbb{E}}_{s\sim\rho_{\boldsymbol{\theta}_k}}\left[\text{KL}(\boldsymbol{\theta}_k\|\boldsymbol{\theta})(s)\right]}, \quad (1)$$

where the first term on the right-hand side is the surrogate objective and the second is the KL penalty term.

This lower bound can be optimized iteratively to generate a sequence of policies with parameters $\{\boldsymbol{\theta}_k\}$ and monotonically improving performance. However, this optimization problem can be difficult to solve and leads to very small policy updates in practice. Instead, TRPO introduces several modifications to this approach to produce a scalable and practical algorithm based on Lemma 1. First, TRPO considers a first order approximation of the surrogate objective and a second order approximation of the KL divergence in (1), yielding the approximate lower bound

$$L(\boldsymbol{\theta}) = \mathbf{g}'(\boldsymbol{\theta} - \boldsymbol{\theta}_k) - \frac{\gamma\epsilon_{\boldsymbol{\theta}}}{(1-\gamma)^2}\sqrt{(\boldsymbol{\theta} - \boldsymbol{\theta}_k)'\mathbf{F}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)}, \quad (2)$$

with

$$\mathbf{g} = \nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta}_k) = \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{(s,a)\sim d_{\boldsymbol{\theta}_k}}\left[A_{\boldsymbol{\theta}_k}(s, a)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}_k}(a \mid s)\right] \quad (3)$$

and

$$\mathbf{F} = \mathop{\mathbb{E}}_{(s,a)\sim d_{\boldsymbol{\theta}_k}}\left[\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}_k}(a \mid s)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}_k}(a \mid s)'\right], \quad (4)$$

where $\mathbf{g}$ is the standard policy gradient determined by the Policy Gradient Theorem (Williams 1992; Sutton et al. 2000) and $\mathbf{F}$ is the average Fisher Information Matrix (Schulman et al. 2015). Note that $\mathbf{g}$ and $\mathbf{F}$ are themselves expectations, so in practice $L(\boldsymbol{\theta})$ is estimated using sample averages $\hat{\mathbf{g}}$ and $\hat{\mathbf{F}}$. TRPO then reformulates the KL penalty term as a trust region constraint to produce policy updates

of meaningful size, resulting in the following optimization problem at each iteration:

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \hat{\mathbf{g}}'(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \\ \text{s.t.} \quad & \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)'\hat{\mathbf{F}}(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \quad \leq \quad \delta_{\text{KL}}, \end{aligned} \tag{5}$$

where the trust region parameter δ_{KL} is chosen based on the desired level of conservatism. The closed-form solution is $\boldsymbol{\theta} = \boldsymbol{\theta}_k + \eta \mathbf{v}$, where $\mathbf{v} = \hat{\mathbf{F}}^{-1}\hat{\mathbf{g}}$ is the update direction and $\eta = \sqrt{2\delta_{\text{KL}}/\mathbf{v}'\hat{\mathbf{F}}\mathbf{v}}$. The update direction cannot be calculated directly in high dimensions, so it is solved approximately by applying a finite number of conjugate gradient steps to $\hat{\mathbf{F}}\mathbf{v} = \hat{\mathbf{g}}$. Finally, a backtracking line search is performed to account for the error introduced by the first and second order approximations.

3 Uncertainty-Aware Trust Region

By replacing expectations with sample-based estimates in the approximate lower bound $L(\boldsymbol{\theta})$, TRPO introduces a potentially significant source of error when the number of samples n used to construct the estimates is small. This finite-sample estimation error can destroy the approximate monotonic improvement argument on which TRPO is based. As a result, TRPO typically uses large amounts of data to generate stable performance.

We first address the error present in the policy gradient estimate. Rather than relying on the high-variance estimate $\hat{\mathbf{g}}$ to approximate $\mathbf{g}$ in $L(\boldsymbol{\theta})$, we instead develop a robust lower bound $L_n(\boldsymbol{\theta})$ that holds for all vectors in an uncertainty set $\mathcal{U}_n$ centered around $\hat{\mathbf{g}}$. If $\mathcal{U}_n$ contains the true policy gradient $\mathbf{g}$, $L_n(\boldsymbol{\theta})$ will be a lower bound to $J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k)$ up to first and second order approximation error.

Consider the policy gradient random vector

$$\boldsymbol{\xi} = \frac{1}{1-\gamma} A_{\boldsymbol{\theta}_k}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}(a \mid s) \in \mathbb{R}^d, \tag{6}$$

where $(s,a) \sim d_{\boldsymbol{\theta}_k}$. Note that $\mathbf{g} = \mathbb{E}[\boldsymbol{\xi}]$ is the true policy gradient as in (3), and $\boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{\xi} - \mathbf{g})(\boldsymbol{\xi} - \mathbf{g})'\right]$ is the true covariance matrix of the policy gradient random vector. We make the following assumption regarding the standardized random vector $\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\xi} - \mathbf{g})$:

Assumption 1. *$\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\xi} - \mathbf{g})$ is a sub-Gaussian random vector with variance proxy σ^2.*

The sub-Gaussian assumption is a reasonable one, and is satisfied by standard assumptions in the literature such as bounded rewards and bounded $\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_k}(a \mid s)$ (Konda and Tsitsiklis 2000; Papini et al. 2018). Using this assumption, we construct $\mathcal{U}_n$ as follows (see the Appendix for more details):

Lemma 2 (Constructing $\mathcal{U}_n$)**.** *Consider $\hat{\boldsymbol{\xi}}_1, \ldots, \hat{\boldsymbol{\xi}}_n$ independent, identically distributed random samples of $\boldsymbol{\xi}$, with $\hat{\mathbf{g}} = \frac{1}{n}\sum_{i=1}^n \hat{\boldsymbol{\xi}}_i$ their sample average. Fix $\alpha \in (0,1)$, and define*

$$\mathcal{U}_n = \left\{ \mathbf{u} \mid (\mathbf{u} - \hat{\mathbf{g}})'\boldsymbol{\Sigma}^{-1}(\mathbf{u} - \hat{\mathbf{g}}) \leq \sigma^2 R_n^2 \right\}, \tag{7}$$

where

$$R_n^2 = \frac{1}{n}\left(d + 2\sqrt{d \log\left(\frac{1}{\alpha}\right)} + 2\log\left(\frac{1}{\alpha}\right) \right). \tag{8}$$

Then, $\mathbf{g} \in \mathcal{U}_n$ with probability at least $1 - \alpha$.

The ellipsoidal uncertainty set $\mathcal{U}_n$ constructed in Lemma 2 has an intuitive structure. It can be seen as a multivariate extension of the standard confidence interval in univariate statistics, where the radius has been calculated to accommodate the more general sub-Gaussian case (Hsu, Kakade, and Zhang 2012). We use this uncertainty set to develop a finite-sample lower bound $L_n(\boldsymbol{\theta})$ on the performance difference between two policies:

Theorem 1 (Finite-Sample Policy Improvement Lower Bound)**.** *Consider two policies $\pi_{\boldsymbol{\theta}_k}$ and $\pi_{\boldsymbol{\theta}}$. Let $\epsilon_{\boldsymbol{\theta}}$ be as defined in Lemma 1. Assume Lemma 2 holds and is used to construct $\mathcal{U}_n$ with confidence $1 - \alpha$. Then, we have that*

$$\begin{aligned} L_n(\boldsymbol{\theta}) = \hat{\mathbf{g}}'(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \\ - \frac{\gamma \epsilon_{\boldsymbol{\theta}}}{(1-\gamma)^2}\sqrt{(\boldsymbol{\theta} - \boldsymbol{\theta}_k)'\mathbf{F}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)} \\ - \sigma R_n \sqrt{(\boldsymbol{\theta} - \boldsymbol{\theta}_k)'\boldsymbol{\Sigma}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)} \end{aligned} \tag{9}$$

is a lower bound for $J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k)$ with probability at least $1 - \alpha$, up to first and second order approximation error.

Proof. Consider a robust (i.e., worst-case with respect to $\mathcal{U}_n$) lower bound of the form

$$\min_{\mathbf{u} \in \mathcal{U}_n} \mathbf{u}'(\boldsymbol{\theta} - \boldsymbol{\theta}_k) - \frac{\gamma \epsilon_{\boldsymbol{\theta}}}{(1-\gamma)^2}\sqrt{(\boldsymbol{\theta} - \boldsymbol{\theta}_k)'\mathbf{F}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)}, \tag{10}$$

where $\mathcal{U}_n$ is defined as in Lemma 2. Note that (10) is a minimization of a linear function of $\mathbf{u}$ subject to a convex quadratic constraint in $\mathbf{u}$. By forming the Lagrangian and applying strong duality, we see that the minimum value of (10) can be written in closed form as $L_n(\boldsymbol{\theta})$. By construction of $\mathcal{U}_n$, $\mathbf{g}$ is a feasible solution to (10) with probability at least $1 - \alpha$. Therefore, $L(\boldsymbol{\theta}) \geq L_n(\boldsymbol{\theta})$ with probability at least $1 - \alpha$. By Lemma 1, $L(\boldsymbol{\theta})$ is a lower bound for $J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k)$ up to first and second order approximation error, which implies that with probability at least $1 - \alpha$ so is $L_n(\boldsymbol{\theta})$. For a more detailed proof, see the Appendix. □

The appearance of an additional penalty term in our robust finite-sample lower bound $L_n(\boldsymbol{\theta})$ motivates the use of the following modified trust region:

Definition 1 (Uncertainty-Aware Trust Region)**.** *For a given δ_{UA}, the uncertainty-aware trust region represents the set of all parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ that satisfy the constraint*

$$\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)'\mathbf{M}(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \leq \delta_{\text{UA}}, \tag{11}$$

where $\mathbf{M} = \mathbf{F} + cR_n^2\boldsymbol{\Sigma}$, $c \geq 0$.

Note that each term in $\mathbf{M}$ accounts for a main source of potential error: the first term controls the approximation error from using on-policy expectations as in TRPO, while the second term controls the finite-sample estimation error from

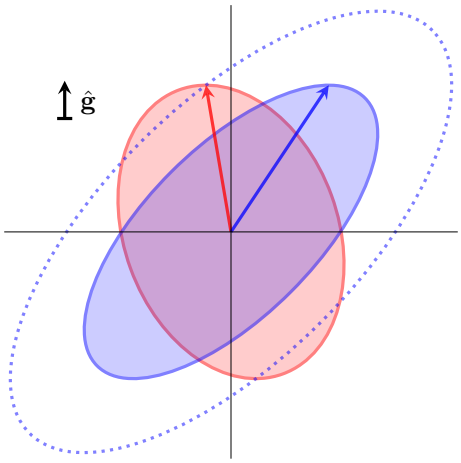

Figure 2: Illustration of trust regions and corresponding policy updates in parameter space for TRPO (red) and UA-TRPO (blue). The dotted blue line represents the smallest uncertainty-aware trust region that contains the update proposed by TRPO. By accounting for the uncertainty present in the policy gradient estimate $\hat{\mathbf{g}}$, UA-TRPO achieves the same level of policy improvement as TRPO with lower total potential error (even though, in this case, UA-TRPO produces a larger policy update than TRPO in terms of KL divergence).

using the policy gradient estimate $\hat{\mathbf{g}}$. The importance of including this second term is illustrated in Figure 2. The resulting trust region adapts to the true noise of the policy gradient random vector through $\boldsymbol{\Sigma}$, as well as the number of samples n used to estimate the policy gradient through the coefficient R_n^2. We include the parameter $c \geq 0$ to control the trade-off between the two terms of $\mathbf{M}$, with $c = 0$ corresponding to standard TRPO (i.e., no penalty for finite-sample estimation error).

This results in a modified policy update based on the optimization problem

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \hat{\mathbf{g}}'(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \\ \text{s.t.} \quad & \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)'\hat{\mathbf{M}}(\boldsymbol{\theta} - \boldsymbol{\theta}_k) \quad \leq \quad \delta_{\text{UA}}, \end{aligned} \tag{12}$$

where $\hat{\mathbf{F}}$ in (5) has been replaced by a sample-based estimate $\hat{\mathbf{M}} = \hat{\mathbf{F}} + cR_n^2\hat{\boldsymbol{\Sigma}}$ of the uncertainty-aware trust region matrix.

4 Uncertainty-Aware Update Direction

The need to use the sample-based estimate $\hat{\mathbf{M}}$ for the trust region in (12) introduces another potential source of error. In particular, because we are estimating a high-dimensional matrix using a limited number of samples, $\hat{\mathbf{M}}$ is unlikely to be full rank. This creates multiple problems when approximating the update direction $\mathbf{v} = \mathbf{M}^{-1}\mathbf{g}$ by solving the system of equations $\hat{\mathbf{M}}\mathbf{v} = \hat{\mathbf{g}}$. First, it is unlikely that this system of equations has an exact solution, so we must consider a least-squares solution instead. Second, the least-squares solution is not unique because the null space of $\hat{\mathbf{M}}$ contains non-zero directions. This second point is particularly important for managing uncertainty, as the null space of $\hat{\mathbf{M}}$ can be interpreted as the directions in parameter space where we have no information on the trust region metric from observed data.

In order to produce a stable, uncertainty-aware policy update, we should restrict our attention to directions in parameter space where an estimate of the trust region metric is available. Mathematically, this means we are interested in finding a least-squares solution to $\hat{\mathbf{M}}\mathbf{v} = \hat{\mathbf{g}}$ that is contained in the row space of $\hat{\mathbf{M}}$ (equivalently, the range of $\hat{\mathbf{M}}$ since the matrix is symmetric). The unique least-squares solution that satisfies this restriction is $\mathbf{v} = \hat{\mathbf{M}}^{+}\hat{\mathbf{g}}$, where $\hat{\mathbf{M}}^{+}$ denotes the Moore-Penrose pseudoinverse of $\hat{\mathbf{M}}$. It is important to note that the standard implementation of TRPO does not produce this update direction in general, leading to unstable and inefficient updates when sample sizes are small.

If $\hat{\mathbf{M}}$ has rank $p < d$, it can be written as $\hat{\mathbf{M}} = \mathbf{U}\mathbf{D}\mathbf{U}'$ where $\mathbf{U} \in \mathbb{R}^{d \times p}$ is an orthonormal eigenbasis for the range of $\hat{\mathbf{M}}$ and $\mathbf{D} \in \mathbb{R}^{p \times p}$ is a diagonal matrix of the corresponding positive eigenvalues. The Moore-Penrose pseudoinverse is $\hat{\mathbf{M}}^{+} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}'$, and the uncertainty-aware update direction can be calculated as $\mathbf{v} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}'\hat{\mathbf{g}}$. Intuitively, this solution is obtained by first restricting $\hat{\mathbf{M}}$ and $\hat{\mathbf{g}}$ to the p-dimensional subspace spanned by the basis $\mathbf{U}$, finding the unique solution to the resulting p-dimensional system of equations, and representing this solution in terms of its coordinates in parameter space.

Unfortunately, standard methods for computing a decomposition of $\hat{\mathbf{M}}$ are computationally intractable in high dimensions. Rather than considering the full range of $\hat{\mathbf{M}}$ spanned by $\mathbf{U}$, we propose to instead restrict policy updates to a low-rank subspace of the range using random projections (Halko, Martinsson, and Tropp 2011). By generating $m \ll d$ random projections, we can efficiently calculate a basis for this subspace and the corresponding uncertainty-aware update direction (see Algorithm 1 for details). Because the subspace is contained in the range of $\hat{\mathbf{M}}$ by construction, we preserve our original goal of restricting policy updates to directions in parameter space where trust region information is available.

Our update method provides the additional benefit of generating a sufficient statistic that can be stored efficiently in memory ($\mathbf{Y}$ in Algorithm 1), which is not the case in TRPO. Because the sufficient statistic can be stored in memory, we can utilize exponential moving averages to produce more stable estimates of the trust region matrix (Wu et al. 2017; Kingma and Ba 2015). This additional source of stability can be implemented through minor modifications to Algorithm 1, which we detail in the Appendix.

5 Algorithm

By applying the uncertainty-aware trust region from Definition 1 and computing an uncertainty-aware update direction via Algorithm 1, we develop a robust policy optimization method that adapts to the uncertainty present in the sample-based estimates of both the policy gradient and

Algorithm 1: Uncertainty-Aware Update Direction

Input: sample-based estimates $\hat{\mathbf{g}} \in \mathbb{R}^d$, $\hat{\mathbf{M}} \in \mathbb{R}^{d \times d}$; random matrix $\mathbf{\Omega} \in \mathbb{R}^{d \times m}$.

Generate m random projections onto the range of $\hat{\mathbf{M}}$:

$$\mathbf{Y} = \hat{\mathbf{M}}\mathbf{\Omega}.$$

Construct basis $\mathbf{Q} \in \mathbb{R}^{d \times \ell}$, $\ell \leq m$, with orthonormal vectors via the singular value decomposition of $\mathbf{Y}$.

Compute projections of $\hat{\mathbf{M}}$, $\hat{\mathbf{g}}$ in the ℓ-dimensional subspace spanned by $\mathbf{Q}$:

$$\widetilde{\mathbf{M}} = \mathbf{Q}'\hat{\mathbf{M}}\mathbf{Q} \in \mathbb{R}^{\ell \times \ell}, \quad \widetilde{\mathbf{g}} = \mathbf{Q}'\hat{\mathbf{g}} \in \mathbb{R}^{\ell}.$$

Construct the eigenvalue decomposition of $\widetilde{\mathbf{M}}$:

$$\widetilde{\mathbf{M}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'.$$

Solve the ℓ-dimensional system $\widetilde{\mathbf{M}}\mathbf{y} = \widetilde{\mathbf{g}}$ for $\mathbf{y} \in \mathbb{R}^{\ell}$:

$$\mathbf{y} = \widetilde{\mathbf{M}}^{-1}\widetilde{\mathbf{g}} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}'\mathbf{Q}'\hat{\mathbf{g}}.$$

Represent $\mathbf{y}$ in parameter space coordinates to obtain an uncertainty-aware update direction $\mathbf{v} \in \mathbb{R}^d$:

$$\mathbf{v} = \mathbf{Q}\mathbf{y} = \mathbf{Q}\mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}'\mathbf{Q}'\hat{\mathbf{g}}.$$

the trust region metric. These important modifications to the standard trust region approach result in our algorithm Uncertainty-Aware Trust Region Policy Optimization (UA-TRPO), which is detailed in Algorithm 2.

6 Experiments

In our experiments, we aim to investigate the robustness and stability of TRPO and UA-TRPO when a limited amount of data is used for each policy update. In order to accomplish this, we perform simulations on several MuJoCo environments (Todorov, Erez, and Tassa 2012) in OpenAI Gym (Brockman et al. 2016). We focus on the locomotion tasks in OpenAI Baselines' (Dhariwal et al. 2017) MuJoCo1M benchmark set (Swimmer-v3, Hopper-v3, HalfCheetah-v3, and Walker2d-v3), all of which have continuous, high-dimensional state and action spaces.

Because we are interested in evaluating the performance of TRPO and UA-TRPO when updates must be made from limited data, we perform policy updates every $1{,}000$ steps in our experiments. The tasks we consider all have a maximum time horizon of $1{,}000$, so our choice of batch size represents as little as one trajectory per policy update. Most implementations of TRPO in the literature make use of larger batch sizes, such as $5{,}000$ (Henderson et al. 2018), $25{,}000$ (Wu et al. 2017), or $50{,}000$ (Duan et al. 2016) steps per policy update. We run each experiment for a total of one million steps, and we consider 50 random seeds.

We represent our policy $\pi_{\boldsymbol{\theta}}$ as a multivariate Gaussian distribution, where the mean action for a given state is parameterized by a neural network with two hidden layers of 64 units each and tanh activations. The standard deviation is parameterized separately, and is independent of the state. This is a commonly used policy structure in deep reinforcement learning with continuous actions (Henderson et al. 2018). The combination of high-dimensional state and action spaces with our neural network policy representation results in policy gradients with dimension d ranging between $4{,}800$ and $5{,}800$.

Algorithm 2: Uncertainty-Aware Trust Region Policy Optimization (UA-TRPO)

Input: initial policy parameterization $\boldsymbol{\theta}_0 \in \mathbb{R}^d$; trust region parameters $\delta_{\text{UA}}, c, \alpha$; random matrix $\mathbf{\Omega} \in \mathbb{R}^{d \times m}$.

for $k = 0, 1, 2, \ldots$ **do**

Collect sample trajectories $\tau_1, \ldots, \tau_n \sim \pi_{\boldsymbol{\theta}_k}$.

Calculate sample-based estimates of the policy gradient $\hat{\mathbf{g}}$ and uncertainty-aware trust region matrix $\hat{\mathbf{M}} = \hat{\mathbf{F}} + cR_n^2\hat{\mathbf{\Sigma}}$.

Use Algorithm 1 to compute an uncertainty-aware update direction $\mathbf{v}$.

Apply the policy update:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta\mathbf{v}, \quad \eta = \sqrt{\frac{2\delta_{\text{UA}}}{\mathbf{v}'\hat{\mathbf{M}}\mathbf{v}}}.$$

end

We use the hyperparameters from Henderson et al. (2018) for our implementation of TRPO, which includes $\delta_{\text{KL}} = 0.01$ (see Equation (5)). For UA-TRPO, we use $\delta_{\text{UA}} = 0.03$, $c = 6\text{e}{-4}$, and $\alpha = 0.05$ for the inputs to Algorithm 2 in all of our experiments. We determined δ_{UA} through cross validation, where the trade-off parameter c was chosen so that on average the KL divergence between consecutive policies is the same as in TRPO to provide a fair comparison. See the Appendix for additional details.

Robustness Comparison. In order to evaluate the robustness of TRPO and UA-TRPO, we consider the conditional value at risk (CVaR) of cumulative reward across our 50 trials. For a given $\kappa \in [0, 1]$, κ-CVaR represents the expected value of the bottom κ quantiles.

First, we consider the CVaR of final performance after one million steps of training. Figure 3 displays the κ-CVaR of final performance for all $\kappa \in [0, 1]$. For small values of κ where κ-CVaR represents a more robust measure of performance, UA-TRPO is comparable to or outperforms TRPO across all environments. In all environments except Walker2d-v3, the final κ-CVaR of UA-TRPO exceeds that of TRPO for almost all values of $\kappa \in [0, 1]$.

As shown in Figure 4, the robustness of UA-TRPO extends beyond final performance. In all environments, UA-TRPO also demonstrates comparable or improved 20%-CVaR throughout the training process. In addition, we see that TRPO actually results in a decrease in 20%-CVaR over

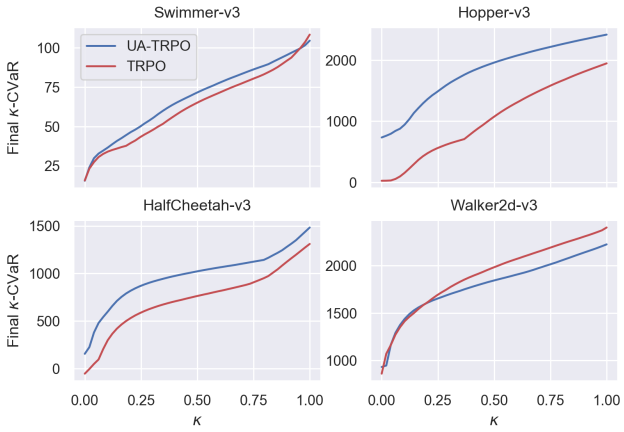

Figure 3: κ-CVaR of final performance for all $\kappa \in [0, 1]$.

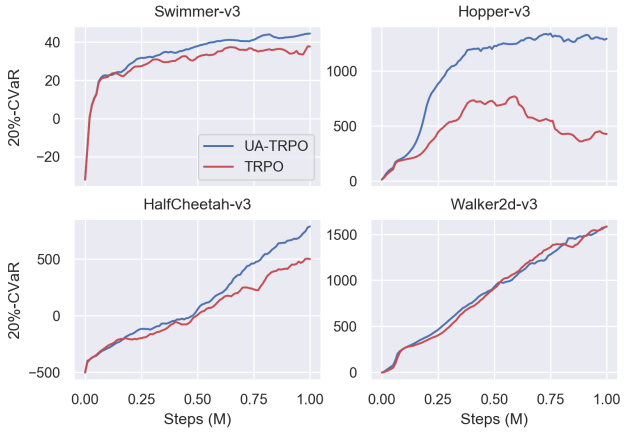

Figure 4: 20%-CVaR of performance throughout training.

time in less stable environments such as Hopper-v3. Clearly, the potential for such instability would be unacceptable in a high-stakes real-world setting.

Average Performance Comparison. Because UA-TRPO is a conservative approach whose primary objective is to guarantee robust policy improvement, it is possible for average performance to suffer in order to achieve robustness. However, the uncertainty-aware trust region automatically adapts to the level of uncertainty present in the problem, which prevents the algorithm from being more conservative than it needs to be. As a result, we find that UA-TRPO is able to improve robustness without sacrificing average performance. We see in Figure 5 that there is no statistically significant difference in average performance between TRPO and UA-TRPO in more stable tasks such as Swimmer-v3, HalfCheetah-v3, and Walker2d-v3. In less stable environments such as Hopper-v3, beyond robustness benefits, UA-TRPO also leads to a statistically significant improvement in average performance.

Adversarial Gradient Noise. We further investigate the robustness of TRPO and UA-TRPO by introducing adversarial noise to the sample-based gradient estimates used to determine policy updates. For each dimension of the gradient, we add noise in the opposite direction of the sample-based estimate. We set the magnitude of this noise to be the standard error of the policy gradient estimate in each dimension, which allows our adversarial noise to mimic the impact of a plausible level of finite-sample estimation error.

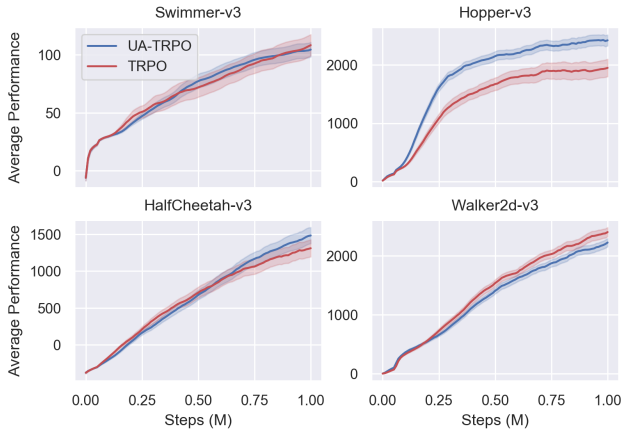

Figure 5: Average performance throughout training. Shading denotes one standard error.

We show the impact of these adversarial gradient perturbations on average performance in Figure 6. As expected, we see a meaningful decrease in performance compared to Figure 5 for both TRPO and UA-TRPO due to the adversarial nature of the noise. However, because the uncertainty-aware trust region penalizes directions with high variance, UA-TRPO is considerably more robust to this noise than TRPO. In the presence of adversarial noise, UA-TRPO demonstrates a statistically significant increase in average performance compared to TRPO for all tasks except Swimmer-v3, including improvements of 98 and 66 percent on HalfCheetah-v3 and Walker2d-v3, respectively.

Quality of Proposed Policy Updates. In order to better understand the performance of TRPO and UA-TRPO with small sample sizes, we analyze the quality of the policy updates proposed by these algorithms. In particular, TRPO is designed to target a specific KL divergence step size determined by δ_{KL}, so we compare the actual KL divergence of the proposed policy update to the KL divergence estimated by the algorithm. Figure 7 shows that almost 40 percent of the updates proposed by TRPO are at least two times larger than desired, and almost 20 percent of the updates are at least three times larger than desired. This major discrepancy between actual and estimated KL divergence is caused by the inability to produce accurate estimates of the trust region matrix from only a small number of samples. TRPO corrects this issue by using a backtracking line search, but the algorithm's strong reliance on this post-optimization process shows that the use of small batch sizes results in inefficient policy updates.

On the other hand, we see in Figure 7 that UA-TRPO pro-

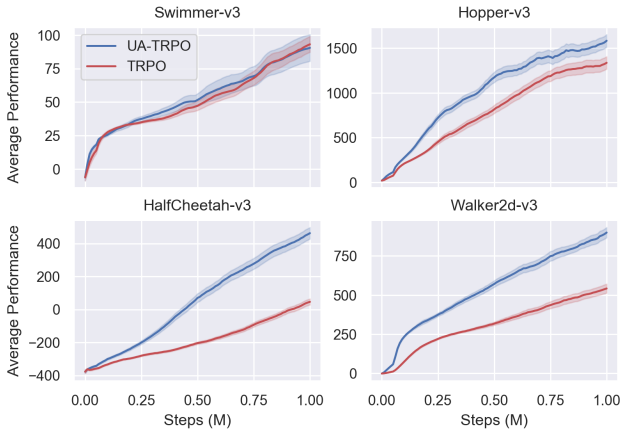

Figure 6: Average performance throughout training with adversarial gradient noise. Shading denotes one standard error.

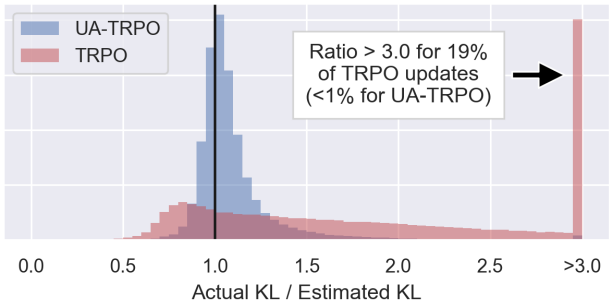

Figure 7: Ratio of actual to estimated KL divergence for proposed policy updates, prior to the application of a backtracking line search. Histogram includes proposed policy updates across all environments and all random seeds.

duces policy updates in line with the algorithm's intended goal (i.e., a ratio near one). This is accomplished by using uncertainty-aware update directions to generate policy updates, which prevent the algorithm from exploiting directions in parameter space where trust region information is not available due to limited data.

7 Related Work

The difficulties of using sample-based estimates in reinforcement learning have been a topic of interest for quite some time. In particular, policy gradient methods are known to suffer from high variance. Variance reduction techniques have been proposed to mitigate this issue, such as the use of baselines and bootstrapping in advantage function estimation (Sutton et al. 2000; Schulman et al. 2016). Recently, there has been renewed interest in variance reduction in the stochastic optimization literature (Johnson and Zhang 2013), and some of these methods have been applied to reinforcement learning (Papini et al. 2018). This line of research is orthogonal to our work, as UA-TRPO adapts to the variance that remains after these techniques have been applied.

Conservative policy optimization approaches such as Conservative Policy Iteration (Kakade and Langford 2002), TRPO (Schulman et al. 2015), and Proximal Policy Optimization (Schulman et al. 2017) consider a lower bound on policy improvement to generate stable policy updates. However, these approaches rely on large batch sizes to control sample-based estimation error. In settings where access to samples may be limited, this approach to reducing estimation error may not be feasible. Li et al. (2011) developed the "knows what it knows" (KWIK) framework for this scenario, which allows an algorithm to choose not to produce an output when uncertainty is high. Several approaches in reinforcement learning can be viewed as applications of this uncertainty-aware framework. Laroche, Trichelair, and Combes (2019) bootstrapped the learned policy with a known baseline policy in areas of the state space where data was limited, while Thomas, Theocharous, and Ghavamzadeh (2015) only produced updates when the value of the new policy exceeded a baseline value with high probability. Our approach is also motivated by the KWIK framework, restricting updates to areas of the parameter space where information is available and producing updates close to the current policy when uncertainty is high.

Other methods have considered adversarial formulations to promote stability in the presence of uncertainty. Rajeswaran et al. (2017) trained a policy with TRPO using adversarially selected trajectories from an ensemble of models, while Pinto et al. (2017) applied TRPO in the presence of an adversary that was jointly trained to destabilize learning. Because both of these methods are motivated by the goal of sim-to-real transfer, they assume that the environment can be altered during training. We introduce an adversary specifically designed to address finite-sample estimation error through a worst-case formulation over $\mathcal{U}_n$, but our method does not require any changes to be made to the underlying environment.

Finally, our approach is related to adaptive learning rate methods such as Adam (Kingma and Ba 2015) that have become popular in stochastic optimization. These methods use estimates of the second moment of the gradient to adapt the step size throughout training. We accomplish this by incorporating $\boldsymbol{\Sigma}$ in our trust region. In fact, adaptive learning rate methods can be interpreted as using an uncentered, diagonal approximation of $\boldsymbol{\Sigma}$ to generate uncertainty-aware updates.

8 Conclusion

We have presented a principled approach to policy optimization in the presence of finite-sample estimation error. We developed techniques that adapt to the uncertainty introduced by sample-based estimates of the policy gradient and trust region metric, resulting in robust and stable updates throughout the learning process. Importantly, our algorithm, UA-TRPO, directly controls estimation error in a scalable and practical way, making it compatible with the use of rich, high-dimensional neural network policy representations. This represents an important step towards developing deep reinforcement learning methods that can be used for real-world decision making tasks where data is limited and stable performance is critical.

Ethics Statement

Reinforcement learning has the potential to improve decision making across many important application areas such as robotics and healthcare, but these techniques will only be useful to society if they can be trusted to produce robust and stable results. Our work makes progress towards accomplishing this goal by addressing major sources of error that currently prevent the real-world adoption of policy optimization algorithms. We do not believe our contributions introduce any ethical issues that could negatively impact society.

Acknowledgments

This research was partially supported by the NSF under grants ECCS-1931600, DMS-1664644, CNS-1645681, and IIS-1914792, by the ONR under grant N00014-19-1-2571, by the NIH under grants R01 GM135930 and UL54 TR004130, by the DOE under grant DE-AR-0001282, by AFOSR under grant FA9550-19-1-0158, by ARPA-E's NEXTCAR program under grant DE-AR0000796, and by the MathWorks.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 22–31. PMLR.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. arXiv preprint. arXiv:1606.01540.

Dhariwal, P.; Hesse, C.; Klimov, O.; Nichol, A.; Plappert, M.; Radford, A.; Schulman, J.; Sidor, S.; Wu, Y.; and Zhokhov, P. 2017. OpenAI Baselines. https://github.com/openai/baselines.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, 1329–1338. PMLR.

Halko, N.; Martinsson, P. G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53(2): 217–288. doi:10.1137/090771806.

Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; and Meger, D. 2018. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 3207–3214. AAAI Press.

Hsu, D.; Kakade, S.; and Zhang, T. 2012. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability* 17. doi:10.1214/ECP.v17-2079.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, 315–323. Curran Associates, Inc.

Kakade, S.; and Langford, J. 2002. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, 267–274. Morgan Kaufmann Publishers Inc.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Konda, V. R.; and Tsitsiklis, J. N. 2000. Actor-critic algorithms. In *Advances in Neural Information Processing Systems 12*, 1008–1014. MIT Press.

Laroche, R.; Trichelair, P.; and Combes, R. T. D. 2019. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 3652–3661. PMLR.

Li, L.; Littman, M. L.; Walsh, T. J.; and Strehl, A. L. 2011. Knows what it knows: A framework for self-aware learning. *Machine Learning* 82: 399–443. doi:10.1007/s10994-010-5225-4.

Papini, M.; Binaghi, D.; Canonaco, G.; Pirotta, M.; and Restelli, M. 2018. Stochastic variance-reduced policy gradient. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 4026–4035. PMLR.

Pinto, L.; Davidson, J.; Sukthankar, R.; and Gupta, A. 2017. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 2817–2826. PMLR.

Rajeswaran, A.; Ghotra, S.; Ravindran, B.; and Levine, S. 2017. EPOpt: Learning robust neural network policies using model ensembles. In *5th International Conference on Learning Representations*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, 1889–1897. PMLR.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M. I.; and Abbeel, P. 2016. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations*.

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

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, 1057–1063. MIT Press.

Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, 2380–2388. PMLR.

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. doi:10.1109/IROS.2012.6386109.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8(3–4): 229–256. doi:10.1007/BF00992696.

Wu, Y.; Mansimov, E.; Grosse, R. B.; Liao, S.; and Ba, J. 2017. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In *Advances in Neural Information Processing Systems 30*, 5279–5288. Curran Associates, Inc.