TOPS: Transition-based VOlatility-controlled Policy Search and its Global Convergence

Liangliang Xu 1  Aiwen Jiang 2  Daoming Lyu 1  Bo Liu 1

Abstract

Risk-averse problems receive far less attention than risk-neutral control problems in reinforcement learning, and existing risk-averse approaches are challenging to deploy to real-world applications. One primary reason is that such risk-averse algorithms often learn from consecutive trajectories with a certain length, which significantly increases the potential danger of causing dangerous failures in practice. This paper proposes Transition-based VOlatility-controlled Policy Search (TOPS), a novel algorithm that solves risk-averse problems by learning from (possibly non-consecutive) transitions instead of only consecutive trajectories. By using an actor-critic scheme with an overparameterized two-layer neural network, our algorithm finds a globally optimal policy at a sublinear rate with proximal policy optimization and natural policy gradient, with effectiveness comparable to the state-of-the-art convergence rate of risk-neutral policy-search methods. The algorithm is evaluated on challenging Mujoco robot simulation tasks under the mean-variance evaluation metric. Both theoretical analysis and experimental results demonstrate a state-of-the-art level of risk-averse policy search methods.

1. Introduction

It has witnessed the successes of reinforcement learning (RL, Sutton and Barto (2018)) in multiple fields and domains (Mnih et al., 2015). However, there are still two concerns with existing RL approaches, which are risk and safety. The first concern, risk, refers to the instability with respect to the uncertainty of future outcomes (Dabney et al., 2020), which are often measured by the variance of the future outcome (e.g., expected cumulative rewards). Controlling risk is necessary in a variety of applications, including financial decision-making (Lai et al., 2011), healthcare (Parker, 2009), and robotics (Majumdar and Pavone, 2020). Most reinforcement learning (RL, Sutton and Barto (2018)) settings are risk-neutral (Mnih et al., 2015; Silver et al., 2016; Vinyals et al., 2019; Wang et al., 2018), meaning that an agent’s goal is merely to learn to maximize the expected utility (cumulative rewards) without caring out the variance of it. Risk-averse RL aims to solve this challenging issue in decision-making and policy search. The second concern, safety, refers to the possibility of triggering hazardous events, which are measured by the probability of entering dangerous states or transitions (García and Fernández, 2015). Unfortunately, safety is not always guaranteed within risk-averse RL. The primary issue is that it is challenging to prevent an agent from triggering hazardous events during learning without limiting exploration efficiency and optimization performance (Cheng et al., 2019). Before we move on, we first introduce the difference between a transition and a trajectory in RL problems. In RL, a transition is a one-step interaction between the agent and the environment, which is defined in detail in Section 2. A trajectory is a series of consecutive uninterrupted transitions. A key observation is that entering dangerous states or transitions are often caused by the agent’s consecutive long-horizon uninterrupted interactions (i.e., trajectory) with the environment (Bisi et al., 2020; Kovács, 2020; Thomas et al., 2021). The possibility of triggering dangerous events would be significantly reduced if the agent does not learn from long-horizon trajectories. However, most existing RL algorithms require learning from long-horizon trajectories, either online interactions or collected in advance in an offline manner. For example, experience replay (ER) (Schaul et al., 2015; Andrychowicz et al., 2017) partially alleviates this problem by enabling learning from past experience. However, most ER still requires past experience to form (possibly long-horizon) trajectories. Therefore, potential safety failure still exists when collecting these (long-horizon) trajectory segments. In this paper, we aim to answer one question: Do volatility-based risk-aware policy gradient algorithms
have global optimality convergence guarantee? Motivated by addressing this question, we propose Transition-based Volatility-controlled Policy Search (TOPS), a risk-averse framework with reward volatility (Bisi et al., 2020) as its risk measurement and establish its global convergence and optimality. This paper makes two major contributions. First, instead of learning from rollouts that are consecutive uninterrupted trajectories of interactions (Zhong et al., 2020; Xu et al., 2021; Xie et al., 2018), our method TOPS does not require to learn from consecutive uninterrupted trajectories. Instead, it can be either consecutive uninterrupted rollouts (also termed trajectories), non-consecutive interrupted interactions (also termed transitions), or a combination of them. This requirement is much milder and more flexible than the consecutive interrupted trajectory requirements. Second, we present a theoretical analysis of the global optimality of the proposed algorithm and prove that TOPS converges to a globally optimal policy at the rate of $1/\sqrt{K}$, where $K$ is the number of iterations. To the best of our knowledge, the closest to this has been achieved by Zhong et al. (2020), which also reaches a converge rate at $1/\sqrt{K}$, using the variance of the cumulative rewards as the risk measure. Our paper is the first to present the global convergence analysis on primal-dual policy search.

The roadmap of the paper is as follows. We introduce the background in Section 2. In Section 3, we formulate the TOPS algorithm. We present the major result on its global convergence in Section 4.2 and provide the theoretical proof in Section C.2. In Section 5, we perform experiments on benchmark domains and compare them with state-of-the-art (SOTA) methods. We discuss some related work in Section 6 and conclude the paper in Section 7.

2. Background

Reinforcement Learning We consider the infinite-horizon discounted Markov Decision Process (MDP) $(S, A, P, r, \gamma)$ with state space $S$, action space $A$, the transition kernel $P: S \times S \times A \rightarrow [0, 1]$, the reward function $r: S \times A \rightarrow \mathbb{R}$, the initial state $S_0 \in S$ and its distribution $\mu_0: S \rightarrow [0, 1]$, and the discounted factor $\gamma \rightarrow (0, 1)$. At time step $t$, given a state $s_t$, an action $a_t$ is taken according to policy $\pi(a_t|s_t): S \times \mathcal{A} \rightarrow [0, 1]$, generating a reward $r_{t+1} = r(s_t, a_t)$ and the next state $s_{t+1}$ based on $p(s_{t+1}|s_t, a_t)$ the reward function is assumed to be deterministic and bounded — a constant $r_{\text{max}} > 0$ exists such that $r_{\text{max}} = \sup_{(s,a) \in S \times A} |r|$. A change in states upon an action $(s, a, r, s')$ is termed a transition, where the state $s'$ is the successive state of the state $s$. A trajectory of length $T$ is a consecutive sequence of transitions $\{(s_t, a_t, r_t, s_t')\}_{t=0}^{T-1}$ over a set of contiguous timestamps, where $\forall t, s_t' = s_{t+1}$. Therefore, the trajectory is also equivalently denoted by $\{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T-1}$. In contrast, a bag of $B$ (possibly non-consecutive) transitions $\{(s_t, a_t, r_t, s_t')\}_{t=0}^{B-1}$ does not necessarily hold that $s_t' \equiv s_{t+1}$. To evaluate the performance of policy $\pi$, we introduce state value function $V_{\pi}: S \rightarrow \mathbb{R}$ and state-action value function $Q_{\pi}: S \times A \rightarrow \mathbb{R}$:

$$V_{\pi}(s) := (1 - \gamma)\mathbb{E}_{a \sim \pi(a|s)} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_t \sim \pi(a|s_t), s_{t+1} \sim \mathcal{P}(s|s_t, a_t) \right],$$

$$Q_{\pi}(s, a) := (1 - \gamma)\mathbb{E}_{a \sim \pi(a|s)} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_t = a, s_{t+1} \sim \mathcal{P}(s|s_t, a_t) \right].$$

Bounded reward implies $|V_{\pi}(s)| \leq r_{\text{max}}$ and $|Q_{\pi}(s, a)| \leq r_{\text{max}} \forall \pi$. Additionally, the advantage function $A_{\pi}: S \times A \rightarrow \mathbb{R}$ of policy $\pi$ is defined as $A_{\pi}(s, a) := Q_{\pi}(s, a) - V_{\pi}(s)$. The normalized state and state action occupancy measure of policy $\pi$ is denoted by $\nu_{\pi}(s)$ and $\sigma_{\pi}(s, a) := \pi(a|s)r_{\pi}(s)$, respectively. Therefore, $\nu_{\pi}(s) := (1 - \gamma)\sum_{t=0}^{\infty} \gamma^t Pr(s_t = s|\mu_0, \pi, \mathcal{P})$ and $\sigma_{\pi}(s, a) := (1 - \gamma)\sum_{t=0}^{\infty} \gamma^t Pr(s_t = s, a_t = a|\mu_0, \pi, \mathcal{P})$, where $Pr(s_t = s|\mu_0, \pi, \mathcal{P})$ is the probability of $s_t = s$ given $\mu_0, \pi, \mathcal{P}$. Finally, the return is defined as $G := \sum_{t=0}^{\infty} \gamma^t r_t$.

Policy Gradient Methods. In the following, we discuss two policy gradient methods, where the policy $\pi_\theta$ is parameterized by the parameter $\theta$. For natural policy gradient (NPG, Kakade (2001)), we first define the Fisher information matrix,

$$F(\theta) := \mathbb{E}_{(s,a) \sim \sigma_{\pi_\theta}} \left[ \nabla_{\theta} \log(\pi_\theta) (\nabla_{\theta} \log(\pi_\theta))^T \right]$$

The update of parameter $\theta$ then takes the form,

$$\theta_{k+1} = \theta_k + \eta_{\text{NPG}} \left( F(\theta_k) \right)^{-1} \nabla J_0(\pi_{\theta_k})$$

where $(F(\theta_{k-1}))^{-1}$ is the inverse of the Fisher information matrix $F(\theta)$ in Eq. (2), and $\eta_{\text{NPG}}$ the learning rate.

In proximal policy optimization (PPO, Schulman et al. (2017)), at the $k$-th iteration the update of policy parameter $\theta$ takes the following form, where $\beta$ is the penalty hyperparameter:

$$\theta_{k+1} = \text{arg max}_{\pi_\theta} \mathbb{E}_{(s,a) \sim \sigma_{\pi_\theta}} \left[ \frac{\pi_\theta}{\pi_{\theta_k}} A_{\theta_k} - \beta \text{KL}(\pi_\theta \| \pi_{\theta_k}) \right].$$

Policy Network with Overparameterized Neural Networks Policy $\pi$ with the two-layer overparameterized neu-
nal network is defined as: for \( \forall (s, a) \in S \times A \),
\[
f((s, a); \theta, b) := \frac{1}{\sqrt{m}} \sum_{i=1}^{m} b_i \text{ReLU}((s, a)^\top [\theta]_i).
\]
Here \((s, a)\) is the input and \( m \) is the width of the network. \( \theta = ([\theta]_1, \ldots, [\theta]_m)^\top \in \mathbb{R}^{m \times d} \) is the input weight matrix in the first layer of the neural network. \( b = (b_1, \ldots, b_m)^\top \in \mathbb{R}^{m \times 1} \) are the output weights in the second layer. We present a block diagram of a overparameterized neural network with Figure 4 in the Appendix A. At the start of training, the parameters \( \theta, b \) are initialized by \( \theta = \Theta_{\text{init}} \in \mathbb{R}^{m \times d} \) (\( \Theta_{\text{init}}[v] \sim \mathcal{N}(0, I_d/d) \)) and \( b_v \sim \text{Unif}([-1, 1]), \forall v \in [m] \), respectively, where \( \mathcal{N} \) denotes Gaussian distribution and Unif denotes uniform distribution. \( f((s, a); \theta, b) \) can be simplified to \( f((s, a); \theta) \) by updating only \( W_v \), during training, and fixing \( b \) as its initialization (Allen-Zhu et al., 2019). We also restrict the possible value of \( \theta \) within the space denoted by \( D = \{ \xi \in \mathbb{R}^{md} : \| \xi - \Theta_{\text{init}} \|_2 \leq Y, Y > 0 \} \). Therefore the policy is defined \( \pi_{\theta}(a|s) \) in the following form:
\[
\pi_{\theta}(a|s) := \frac{\exp \left( \tau f((s, a); \theta) \right)}{\sum_{a' \in A} \exp \left( \tau f((s, a'); \theta) \right)}, \quad \forall (s, a) \in S \times A
\]
\( \tau \) is the temperature parameter. Furthermore, the feature mapping of a two-layer neural network \( f((s, a); \theta) \) is defined as, \( \phi_{\theta} := ([\phi_{\theta}]_1^\top, \ldots, [\phi_{\theta}]_m^\top)^\top \), where \( [\phi_{\theta}]_u^\top = \frac{b_u}{\sqrt{m}} \text{ReLU}((s, a)^\top [\theta]_u) \), \( \forall u \in [m] \). By Eq. (2), it holds that \( f((s, a); \theta) = \phi(s, a)^\top \theta \) and \( \nabla_{\theta} f((s, a); \theta) = \phi(s, a) \) (Wang et al., 2019). We assume that there exists a constant \( M > 0 \) such that,
\[
E_{(s, a) \sim \text{init}} \left[ \sup_{(s, a) \in S \times A} \left| \phi((s, a)^\top \Theta_{\text{init}}) \right|^2 \right] \leq M^2.
\]

### Mean-Variance & Mean-Volatility RL

In a variance-constraint problem with the variance of the total reward, the objective can be formulated as,
\[
\max_{\pi} J(\pi), \quad \text{subject to } \mathbb{V}(G) \leq Y
\]
where \( J(\pi) := \mathbb{E}_{(s, a) \sim \sigma_\pi}[G] = \frac{1}{1-\gamma} \mathbb{E}_{(s, a) \sim \sigma_\pi}[r] \) is the expected return, \( \mathbb{V}(\cdot) \) is the variance of a random variable and \( Y > 0 \) is the upper bound for this variance. The constrained formulation in Eq. (2) is NP-hard (Sobel, 1982), and in reality, the relaxed formulation \( J_\lambda^G(\pi) \) defined in Eq. (2) is often solved instead (Di Castro et al., 2012; L.A. and Ghavamzadeh, 2013; Xie et al., 2018) as follows, where \( \lambda \) is called variance-controlling parameter.
\[
J_\lambda^G(\pi) := \mathbb{E}[G] - \lambda \mathbb{V}(G) = \mathbb{E}[G] - \lambda \mathbb{E}[G^2] + \lambda (\mathbb{E}[G])^2
\]

Meanwhile, (Bisi et al., 2020) proposed a reward-volatility risk measure. \textit{Volatility} is defined as the variance of per-step reward — per-step reward \( R \) is a discrete random variable with a probability mass function of \( p(R = x) = \sum_{s, a} \pi_s(s, a) \mathbb{I}\{r = x\} \), where \( \mathbb{I}\{\cdot\} \) is the indicator function. It is easy to see that \( \mathbb{E}[R] = (1 - \gamma)J(\pi) \) (Zhang et al., 2020). \( \mathbb{V}(R) \) is the variance of \( R \). Likewise, \( J_\lambda(\pi) \) is proposed as a counterpart of Eq. 2 in the sequel, which is defined with respect to \( R \).
\[
J_\lambda(\pi) := \mathbb{E}[R] - \lambda \mathbb{V}(R) = \mathbb{E}[R] - \lambda \mathbb{E}[R^2] + \lambda (\mathbb{E}[R])^2
\]

We first give the following lemma, which is based on Lemma 1 in (Bisi et al., 2020).

**Lemma 1.** Given \( \lambda \geq 0 \), \( J_{\lambda \gamma}(\pi) \) is a lower-bound of \( J_{\lambda}(\pi) \), i.e., \( J_{\lambda \gamma}(\pi) \leq J_{\lambda}(\pi) \).

**Proof.** First, we have \( \mathbb{E}[G] = \mathbb{E}[R] \), i.e., the per-step reward \( R \) is an unbiased estimator of the cumulative reward \( G \). Second, it is proved that \( \mathbb{V}(G) \leq \frac{\mathbb{V}(R)}{1-\gamma^2} \) (Bisi et al., 2020). Given \( \lambda \geq 0 \), summing up \( \mathbb{E}[G] \) with \( -\lambda \mathbb{V}(G) \) (resp. summing up \( \mathbb{E}[R] \) with \( -\lambda \mathbb{E}[R^2] \)) completes the proof. \( \square \)

Given Lemma 1, maximizing \( J_\lambda^G(\pi) \) can be reduced to maximizing its lower bound \( J_{\lambda \gamma}(\pi) \). There are several advantages of optimizing \( J_\lambda(\pi) \). Compared to optimizing \( \mathbb{V}(G) \), optimizing \( \mathbb{V}(R) \) is computationally easier (Zhang et al., 2020). (Bisi et al., 2020) argue that \( \mathbb{V}(R) \) is better at capturing short-term risk and leads to smoother trajectories that avoid possible “shocks” caused by long-horizon trajectories.

### 3. Algorithm Formulation

In this section, we present our risk-averse policy-search algorithm. In particular, we use (i) reward volatility to construct the mean-volatility objective, which circumvents the long-horizon reward issue and avoids large variances, and (ii) overparameterized neural network (Cao and Gu, 2019) as the neural network architecture of the actor and the critic to facilitate global convergence analysis.

#### 3.1. Augmented MDP

As Bisi et al. (2020) shows, reward volatility has advantages over mean-variance methods, including a smoother trajectory and much-reduced variance. Therefore, in our paper, we choose volatility as the risk measurement. Compared with the conventional mean-variance objective, the mean-volatility objective function enables the agent to learn from transitions instead of trajectories and therefore greatly
We now introduce the augmented MDP, with the augmented TOPS illustrated in Figure 1, where there are three sets of parameters to update, i.e., $\theta$ for the actor, $\omega$ for the critic, and the auxiliary variable $y$. Note that the MVPI framework allows incorporating any off-the-shelf policy optimization methods as pointed out by (Zhang et al., 2021). Since most global optimality analysis literature is based on NPG and PPO (Cai et al., 2020; Wang et al., 2019; Liu et al., 2019; Zhong et al., 2020), we also use NPG and PPO as inner policy search algorithms for a fair comparison.

$y$ update: Since Eq. (3.1) is quadratic in $y$, to update $y$ in each iteration, we have $y_k = (1 - \gamma) J(\pi_k)$. However, we do not have direct access to the exact value of $J(\pi_k)$. As an alternative, we estimate this value with a sample average in the $k$-th iteration $\hat{y}_k := (1 - \gamma) B \sum_{i=1}^{B} r_i$, as an estimator of $y_k$, where $B$ is the sample (batch) size. To avoid double-sampling, we resort to augmented MDP. First, note that it holds $(E[R])^2 = \max_{y \in \mathbb{R}} (2E[R] - y^2)$. Then the optimization objective transforms into:

$$\max_{\pi, y} \pi_\lambda(y) = E_{(s,a) \sim \tau^{s,a}}(r - \lambda r^2 + 2\lambda ry) - \lambda y^2$$

We now introduce the augmented reward, with the augmented reward defined as follows:

$$\tilde{r} = r - \lambda r^2 + 2\lambda ry$$

In the remainder of this paper, we refer to this new MDP as the augmented MDP $\tilde{M} = \{S, A, P, \tilde{r}, \gamma\}$, and denote corresponding terms by the `tilde` sign — for example, the associated state value function and state-action value function are $\tilde{V}_\pi(s)$ and $\tilde{Q}_\pi(s, a)$. We solve Eq. (3.1) by maximizing $y$ and $\pi$ of the augmented MDP in two steps iteratively.

### 3.2. Proposed Algorithms

We present TOPS in Algorithm 1. A block diagram of TOPS is illustrated in Figure 1, where there are three sets of parameters to update, i.e., $\theta$ for the actor, $\omega$ for the critic, and the auxiliary variable $y$. Note that the MVPI framework allows incorporating any off-the-shelf policy optimization methods as pointed out by (Zhang et al., 2021). Since most global optimality analysis literature is based on NPG and PPO (Cai et al., 2020; Wang et al., 2019; Liu et al., 2019; Zhong et al., 2020), we also use NPG and PPO as inner policy search algorithms for a fair comparison.

$\theta$ update: We update $\pi_\theta$ with an actor-critic scheme, particularly NPG and PPO. NPG and PPO are the two most widely used policy gradient methods. According to empirical studies (Weng et al., 2020), PPO usually achieves the SOTA performance among on-policy policy gradient methods, and NPG has the advantage of easy hyperparameter tuning compared with PPO. On the other hand, NPG and PPO’s global convergences under the risk-neutral setting have been studied intensively and can show good results (Agarwal et al., 2020; Wang et al., 2019). Therefore, PPO and PNG are used as the inner actor algorithm of the TOPS framework to make fair comparisons with existing approaches. Additionally, overparameterized neural networks are widely used in proving global convergence of gradient-based methods under the risk-neutral setting, which show impressive results (Liu et al., 2019; Wang et al., 2019; Fu et al., 2020). The capability of a gradient-based neural network method to reach the global optimum in an overparameterization setting is explained in theory (Cao and Gu, 2019). Therefore, we parameterize the policy in the paper with a two-layer overparameterized neural network. We first introduce the actor part of the two methods, respectively.

**θ Update for Neural NPG** Per the update rule for NPG in Eq. (2), we need to estimate the natural policy gradient $(F(\theta_k))^{-1} \nabla_\theta J(\pi_\theta)$.

$$\hat{\nabla}_\theta J(\pi_{\theta_k}) := \frac{\tau_k}{B} \sum_{t=1}^{B} Q_{\omega_k}(s_t, a_t) \phi_{\theta_k}(s_t, a_t)$$

are unbiased estimations of $\nabla_\theta J(\pi_\theta)$ and $F(\theta_k)$ respectively, with the help of feature mapping and a bag of transitions of
size $B$, $\theta$ is, therefore, updated as,

$$
\tau_{k+1} = \tau_k + \eta_{NPG},
\theta_{k+1} = (\tau_k \theta_k + 
\eta_{NPG} \arg \min_{\xi \in \mathcal{D}} \| \tilde{F}(\theta_k)\xi - \tau_k \nabla_\theta J(\pi_{\theta_k})\|_2) / \tau_{k+1}
$$

\textbf{\theta Update for Neural PPO} Given the update rule of PPO in Eq. (2), the PPO’s objective function $L(\theta)$ can be rewritten as,

$$
L(\theta) := \mathbb{E}_{s \sim \nu_{\pi}} \left[ \mathbb{E}_{a \sim \pi} [Q_{\omega_k} - \beta KL(\pi_{\theta_k} || \pi_{\theta_k})] \right]
$$

With energy-based policy $\pi \propto \exp\{\tau^{-1} f\}$ from Eq. (2), the solution to the subproblem $\hat{\pi}_{k+1} = \arg \max_\pi L(\theta)$ can be obtained by solving the following (Liu et al., 2019):

$$
\theta_{k+1} = \arg \min_{\theta \in \mathcal{D}} \mathbb{E}_{(s,a) \sim \nu} [(f_\theta - \tau_{k+1}(\beta^{-1} Q_{\omega_k} + 
\tau_{k}^{-1} f_\theta))]^2)
$$

The stochastic gradient method can be used to solve Eq. (3.2). We include details in the Appendix.

\textbf{ω update} To estimate the state-action function value of the augmented MDP $Q_\pi$, a critic network parameterized by $\omega$ is constructed, denoted as $Q_\omega$. For simplicity, the critic uses the same two-layer neural network architecture as the actor defined in Eq. (2). The critic network is parameterized with a different set of parameters $\omega = (\omega_1, \ldots, \omega_m)^T \in \mathbb{R}^{md}$, denoted by $f((s, a); \omega)$. For simplicity, we then learn $Q_\omega$ by applying the semi-gradient TD method. Other approaches, such as the gradient TD (GTD) algorithm family (Sutton et al., 2009), can also be applied. For each iteration $t$ of the TD update,

$$
\omega_{t+1} = \omega_t - \eta_{TD} \left( \tilde{Q}_{\omega_k}(s, a)

- (1 - \gamma) \tilde{r}(s, a) - \gamma Q_{\omega_k}(s', a') \right) \nabla_\omega Q_{\omega_k}(s, a),
$$

where $\eta_{TD}$ is the learning rate for TD update.

\section{4. Theoretical Analysis}

Although NPG and PPO’s global convergences under the risk-neutral setting show prominent result (Agarwal et al., 2020; Wang et al., 2019), the techniques used by these methods only apply to the primal constrained-MDP case. It remains challenging to apply them to the analysis of the primal-dual case as in our augmented MDP. For example, Lemma (5.2) of Liu et al. (2019) is a critical part in the error bounding of risk-neutral PPO’s policy improvement. But since we deploy the augmented MDP instead of the original MDP, we need to analyze the error bound for additional terms, including terms like $\gamma * Q_\pi$, which are hard to tackle. In this section, we establish the global convergence rate of TOPS with both NPG and PPO.

\textbf{Algorithm 1} TOPS: Transition-based VOlatility-controlled Policy Search

1: \textbf{Input:} number of iteration $K$, number of neural TD $T_{TD}$, learning rate for natural policy gradient (resp. PPO) and neural TD $\eta_{NPG}$ (resp. $\eta_{PPO}$) and $\eta_{TD}$, temperature parameters $\{\tau_k\}_{K}^{1}$. 
2: \textbf{Initialization:} Initialize policy network $f((s,a); \theta, \omega)$ with $b_\theta \sim Unif\{-1,1\}$ and $[\Theta_{init}]_\theta \sim \mathcal{N}(0, \hat{d}_d/d), \forall \theta \in [d]$. Set $D = \{\xi \in \mathbb{R}^{md} : \|\xi - \Theta_{init}\|_2 \leq \gamma\}, \theta_1 = \Theta_{init}$, and $\tau_1 = 1$.
3: for $k = 1, \ldots, K$ do 
4: Sample a batch of transitions $\{s_t, a_t, r_t, s'_{t+1}\}_{t=1}^{B}$ following current policy with size of $B$.
5: $y = \frac{1}{B} \sum_{t=1}^{B} r_t$.
6: for $t = 1, \ldots, B$ do 
7: $\tilde{r}_t = r_t - \lambda r_{t-1} + 2 \lambda r_{t-1} a'_t \sim \pi(a|s'_t)$
8: end for
9: \textbf{Q-value update:} initialize the parameters of Q value function network $\beta$ and $\omega_k$ with the similar setting in Line 2, and update $\omega_k$ according to Eq. (3.2).
10: Output $Q_{\omega_k} = Q_{\omega(T_{TD})}$.
11: if select NPG update then
12: update $\theta_k$ according to Eq. (3.2).
13: else if select PPO update then
14: update $\theta_k$ according to Eq. (3.2).
15: end if
16: end for
17: Output: $\pi_{\theta_K}$

\section{4.1. Assumptions}

We first imposing regularity condition assumptions on the action-value function $Q_\pi$ and state stationary distribution $\nu_\sigma$. These assumptions are common in the literature on TD analysis with a neural network approximation (Cai et al., 2020; Wang et al., 2019; Liu et al., 2019; Zhong et al., 2020).

\textbf{Assumption 1.} (Variance upper bound) (Wang et al., 2019) Let $D = \{\alpha \in \mathbb{R}^{md} : \|\alpha - \Theta_{init}\|_2 \leq \gamma\}$. For all $k \in [K]$, We assume that for all $k \in [K]$, there exists an absolute constant $\sigma_\xi > 0$ such that,

$$
\mathbb{E}_{(s,a) \sim \pi_{\omega_k}} [\| \xi_k(\delta_k) \|_2^2] \leq \tau_k^2 \sigma_\xi^2 / B,
\mathbb{E}_{(s,a) \sim \pi_{\omega_k}} [\| \xi_k(\omega_k) \|_2^2] \leq \tau_k^4 \sigma_\xi^2 / B.
$$

where $\delta_k = \arg \min_{\alpha \in \mathcal{D}} \| \tilde{F}(\theta_k)\alpha - \tau_k \nabla_\theta J(\pi_{\theta_k})\|_2$ and $\xi_k(\alpha) = \tilde{F}(\theta_k)\alpha - \tau_k \nabla_\theta J(\pi_{\theta_k}) - \mathbb{E}[\tilde{F}(\theta_k)\alpha - \tau_k \nabla_\theta J(\pi_{\theta_k}) ]$.

We impose a regularity condition on visitation measures and stationary distributions in the sequel, respectively.

\textbf{Assumption 2.} (Upper bounded concentrability coefficient) (Wang et al., 2019) $\nu^*$ and $\sigma^*$ are denoted as the state and
state-action visitation measures corresponding to the global optimum $\pi^*$. For all $k \in [K]$, we define the following terms:

$$
\varphi_k = \mathbb{E}_{(s,a) \sim \sigma_{\pi_k}} \left[ \left( \frac{d\sigma^*}{d\sigma_{\pi_k}} \right)^2 \right]^{1/2},
$$

$$
\psi_k = \mathbb{E}_{s \sim \nu_{\pi_k}} \left[ \left( \frac{d\nu^*}{d\nu_{\pi_k}} \right)^2 \right]^{1/2},
$$

$$
\varphi'_k = \mathbb{E}_{(s,a) \sim \sigma'_{\pi_k}} \left[ \left( \frac{d\sigma^*}{d\sigma'_{\pi_k}} \right)^2 \right]^{1/2},
$$

$$
\psi'_k = \mathbb{E}_{s \sim \nu'_{\pi_k}} \left[ \left( \frac{d\nu^*}{d\nu'_{\pi_k}} \right)^2 \right]^{1/2}.
$$

We assume that $\varphi_k, \psi_k, \varphi'_k, \psi'_k$ are uniformly upper bounded by an absolute constant $c_0 > 0$.

$\sigma_{\pi_k}$ and $\nu_{\pi_k}$ are state-action and state stationary distribution. $\varphi_k, \psi_k, \varphi'_k, \psi'_k$ are the concentration coefficients, which reflects how much the starting state and state-action distribution diverge from the state and state-action distribution under the optimal policy (Munos, 2007). Assumption 2 impose a upper bound on such divergence. This regularity condition is commonly used in the literature (Munos and Szepesvári, 2008; Antos et al., 2008; massoud Farahm et al., 2016; Yang and Wang, 2020; Wang et al., 2019). More over, we define $\varphi_k^* = \mathbb{E}_{(s,a) \sim \sigma_{\pi}} \left[ \left( \frac{d\sigma^*}{d\sigma_{\pi}} - \frac{d\sigma^*}{d\pi_{\pi_k}} \right)^2 \right]^{1/2}$, $\psi_k^* = \mathbb{E}_{s \sim \nu_{\pi}} \left[ \left( \frac{d\nu^*}{d\nu_{\pi}} - \frac{d\nu^*}{d\nu_{\pi_k}} \right)^2 \right]^{1/2}$.

4.2. Major Theoretical results

In the following, we present the major theoretical results, i.e., the global optimality and convergence rate of TOPS with neural PPO. We define the optimality gap $\min_{k \in [K]} (J^\pi_{\lambda_k}(\pi^*) - J^\pi_{\lambda_k}(\pi_k))$, where $J^\pi_{\lambda_k}(\pi^*)$ and $J^\pi_{\lambda_k}(\pi_k)$ represent the risk-averse objective under global optimal policy $\pi^* := \arg \max_{\pi} J_{\lambda}(\pi)$ and TOPS-generated policy at the $k$-th iteration $\pi_k$, respectively.

**Theorem 1.** (Global Optimality and Rate of Convergence for neural PPO) We set the learning rate of PPO $\eta_{\text{PPO}} = 1/\sqrt{K}$, the learning rate of TD update $\eta_{\text{TD}} = \min\{(1 - \gamma)/3(1 + \gamma)^2, 1/\sqrt{TD}\}$, and the temperature parameters $\tau_k = (k - 1)\eta_{\text{PPO}}$. Under Assumptions 3-4, with a probability of $1 - \delta$, we have

$$
\min_{k \in [K]} (J^\pi_{\lambda_k}(\pi^*) - J^\pi_{\lambda_k}(\pi_k)) \leq \frac{1}{(1 - \gamma)\sqrt{K}} \left( \log |A| + 9\Upsilon^2 + M^2 + 4\epsilon_3 M (1 - \gamma)^2 \lambda \right) + \frac{1}{K} \sum_{k=1}^{K} (\epsilon_k),
$$

where $\epsilon_k(K) = \sqrt{8c_0 \Upsilon^1/\tau_k^{1/2} B^{-1/4}} + \mathcal{O}\left((\tau_{k+1}K^{1/2} + 1) \Upsilon^3/2m^{-1/4} + \Upsilon^5/4m^{-1/8}\right) + c_0 \mathcal{O}\left(\Upsilon^3 m^{-1/2} \log(1/\delta) + \Upsilon^5/2m^{-1/4} \sqrt{\log(1/\delta)}\right) + \Upsilon^2 m^{-1/4} + \Upsilon^2 T_{\text{TD}}^{-1/2} + \Upsilon$.

Similarly, we present TOPS global optimality and convergence rate with neural PPO.

**Theorem 2.** (Global Optimality and Rate of Convergence on neural PPO) We set the learning rate of PPO $\eta_{\text{PPO}} = \min\{(1 - \gamma)/3(1 + \gamma)^2/\sqrt{TD}\}$, the learning rate of TD update $\eta_{\text{TD}} = \min\{(1 - \gamma)/3(1 + \gamma)^2, 1/\sqrt{TD}\}$, and $\beta_0 := \beta/\sqrt{K}$. Under Assumptions 3-4, we have, with a probability of $1 - \delta$,

$$
\min_{k \in [K]} (J^\pi_{\lambda_k}(\pi^*) - J^\pi_{\lambda_k}(\pi_k)) \leq \beta_0^2 \log |A| + U + \beta_0^2 \sum_{k=1}^{K} (\epsilon_k + \epsilon_k') + 4\epsilon_3 m \beta_0 (1 - \gamma)^2 \frac{1}{(1 - \gamma)\sqrt{K}},
$$

where $\epsilon_k = \epsilon_k' = \mathcal{O}\left(\Upsilon^3 m^{-1/2} \log(1/\delta) + \Upsilon^5/2m^{-1/4} + \Upsilon^2 T_{\text{TD}}^{-1/2} + \Upsilon\right)$,

$$
\epsilon_k = \tau_{k+1}^{-1} \epsilon_k + \beta_1 \epsilon_k' \psi_k + \epsilon^*_k = |A| \tau_{k+1}^{-1} \psi_k^*,
$$

$$
U = 2\mathbb{E}_{s \sim \nu_{\pi^*}} \left[ \max_{a \in A} (\hat{Q}_{\omega_0})_a^2 \right] + 2\Upsilon^2.
$$

**Remark 1.** Theorem 1 and 2 show the upper bound of the optimality gap $\min_{k \in [K]} (J^\pi_{\lambda_k}(\pi^*) - J^\pi_{\lambda_k}(\pi_k)) \sim O\left(\frac{1}{K}\right)$, where $K$ is the maximum number of updates. It reflects how close the policy produced by TOPS can achieve to the global optimal policy.

From Theorem 1 and 2, we can conclude that our risk-averse algorithm TOPS with PPO version of the actor both converge to the global optimal policy at a $O(1/\sqrt{K})$ rate.

5. Experiments

One of the real-world domains of interest in risk-averse RL is robotics. Robots are expected to achieve consistent...
TOPS: Transition-based VOlatility-controlled Policy Search and its Global Convergence

Figure 2. Training progress of TOPS-NPG and baseline algorithms.

Figure 3. Training progress of TOPS-PPO and baseline algorithms.

performance while avoiding failures that lead to dangerous results. Therefore, we perform the experiments for TOPS on the Mujoco robot manipulation benchmark tasks from OpenAI gym (Brockman et al., 2016). We conduct our experiments in an online learning setting and use several recent risk-averse RL methods as baselines. Namely, the mean-variance policy optimization (MVP) (Xie et al., 2018), mean-variance policy iteration (MVPI) (Zhang et al., 2020), and variance-constrained actor-critic (VARAC) (Zhong et al., 2020).

For MVP, since the algorithm uses coordinate gradient at each step, it does not have a PPO or NPG version, and therefore we only report its learning curve in Figure 2. For MVPI, we use its on-policy version for the experiment. As it works with any off-the-shelf policy search method, we implement NPG and PPO with two-layer overparameterized
neural networks as its policy search component.

We set $\lambda = 1$ and run each algorithm for $10^6$ steps and evaluate the algorithm every $10^4$ steps for 20 episodes. We report the mean of the algorithms’ returns against the training steps in Figure 2 & 3. All curves are averaged over 10 independent runs and use shaded areas to indicate standard errors. The experiment’s details are provided in the Appendix. All experiments’ parameters are tuned through rigorous grid search. There are several interesting findings of our experiments. First, on every single task, the order of performance level tends to be identical, regardless of the base method (NPG or PPO). For example, in Walker2d domain, the order of performance level is (from the best to the worst) in the upper-left subfigure of Figure 2 is TOPS-NPG, VARAC-NPG, and MVPI-NPG. This is consistent with the results reported in the upper-left subfigure of Figure 3, where the order is TOPS-PPO, VARAC-PPO, and MVPI-PPO. Compared to other methods, MVP performs poorly in all tested domains, which indicates it may not suit the tasks. Second, the results show that TOPS outperforms other baselines in most of the testbeds. In particular, TOPS outperforms other methods with a larger margin on Walker2d, Hopper and Walker2d, as seen from Figure 2 and Figure 3. Overall, these results demonstrate that our TOPS algorithm can achieve state-of-art risk-averse performance on the challenging robot simulator testbeds.

6. Discussion with related work

First, we distinguish our method with Xie et al. (2018), Bisi et al. (2020), and Zhang et al. (2020). Xie et al. (2018) utilize the variance of the cumulative rewards. However, this method’s theoretical analysis is limited to the sample complexity of episodic average-reward MDP, and the choice of solvers is restricted. Both Bisi et al. (2020) and Zhang et al. (2020) use the variance of the per-step reward, which introduces a policy-dependent-reward issue. Bisi et al. (2020) solve this directly, which is much more difficult than normal MDP due to the lack of tools, and this approach also requires double-sampling. Zhang et al. (2020) avoid these issues by proposing an augmented MDP in a flexible framework that can apply any off-the-shelf policy evaluation and control method. Our analysis is inspired by Zhang et al. (2020), however, our approach differs from theirs in the method: In each iteration of $y$, they keep updating $\pi$ until the objective function is maximized, while we only update $\pi$ once, which enables the training to be faster. None of these papers provides theoretical proof of global convergence.

Second, we compare our proof technique with those of Wang et al. (2019); Liu et al. (2019); Zhong et al. (2020); Xu et al. (2021). These papers follow the same logic flow and share a similar method in the first part of the analysis, but the second parts are different because each works on a different policy gradient method. Our paper adopts the methods of Wang et al. (2019) and Liu et al. (2019) for neural NPG and neural PPO, respectively, and develops a method for the Q-function value based on the two. Compared with Wang et al. (2019) and Liu et al. (2019), our results of convergence are in probability rather than expectation because we utilize different techniques when we characterize certain error bounds.

The closest work to ours is Zhong et al. (2020), which utilizes the variance of the cumulative rewards and therefore needs to learn from consecutive trajectories instead of non-consecutive transitions. Therefore, a corresponding disadvantage is that it requires an extra critic component to estimate the cumulative rewards, which is unnecessary for TOPS. However, in their proof, Eq.(4.15) does not hold, which leads to an invalid core conclusion regarding their Eq.(4.17), an essential part of the proof. It remains unclear if this can be fixed or not, and the theoretical analysis after that remains questionable. The details are mentioned in Appendix C.1. Therefore, our analysis is the first to prove the global convergence in the primal-dual policy search. Moreover, Zhong et al. (2020) do not provide empirical results in their paper. The most recent work along this research line is Xu et al. (2021), but their analysis can be only applied to a specific variant of NPG instead of the general NPG formulation. In contrast, our analysis applies to the general NPG and PPO algorithms, with more details. Other related work includes Zou et al. (2020), Xu et al. (2018), and Bhandari and Russo (2019).

7. Conclusion

This paper proposes TOPS, a risk-averse RL framework utilizing reward volatility that incorporates overparameterized two-layer neural networks. Theoretical analysis for both neural NPG and neural PPO with two-layer overparameterized neural networks are presented to show that TOPS can find the global optimality at an $O(1/\sqrt{K})$ converge rate. We also demonstrate the empirical success of TOPS in Mujoco robot simulation domains.

References

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift, 2020.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization, 2019.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hind-
sight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.

András Antos, Csaba Szepesvári, and Rémi Munos. Fitted q-iteration in continuous action-space mdps. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

Lorenzo Bisi, Luca SABBioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4583–4589, 7 2020. Special Track on AI in FinTech.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Qi Cai, Zhuoran Yang, Jason D. Lee, and Zhaoran Wang. Neural temporal-difference and q-learning provably converge to global optima, 2020.

Yuan Cao and Quanquan Gu. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks, 2019.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3387–3395, 2019.

Imre Csiszár and János Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. A distributional code for value in dopamine-based reinforcement learning. *Nature*, 577 (7792):671–675, 2020.

Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. *arXiv preprint arXiv:1206.6404*, 2012.

Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Single-timescale actor-critic provably finds globally optimal policy. *arXiv preprint arXiv:2008.00483*, 2020.
Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, February 2015. ISSN 00280836.

Rémi Munos. Performance bounds in Lp-norm for approximate value iteration. SIAM journal on control and optimization, 46(2):541–561, 2007.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. Journal of Machine Learning Research, 9(27):818–822, 2009.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural Information Processing Systems, volume 21. Curran Associates, Inc., 2009.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. arXiv preprint arXiv:1511.05952, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewal, John Nham, Nal Calkhoun, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Darius Hassabis. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587):484–489, January 2016. doi: 10.1038/nature16961.

Matthew J Sobel. The variance of discounted markov decision processes. Journal of Applied Probability, 19(4):794–802, 1982.

Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 993–1000, 2009.

Garrett Thomas, Yuping Luo, and Tengyu Ma. Safe reinforcement learning by imagining the near future. Advances in Neural Information Processing Systems, 34, 2021.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michael B. Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature, 575(7782):350–354, 2019.

L. Wang, Q. Cai, Z. Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence, 2019.

M. Wang, E. X Fang, and H Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. Mathematical Programming, 161(1-2):419–449, 2017.

William Yang Wang, Jiwei Li, and Xiaodong He. Deep reinforcement learning for nlp. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pages 19–21, 2018.

jiayi Weng, Alexis Duburcq, Kaichao You, and Huayu Chen. Mujoco benchmark, 2020. URL https://tianshou.readthedocs.io/en/master/tutorials/benchmark.html.

Tengyang Xie, Bo Liu, Yangyang Xu, Mohammad Ghavamzadeh, Yinlam Chow, Daoming Lyu, and Dae-sub Yoon. A block coordinate ascent algorithm for mean-variance optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, 2018.

Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of langevin dynamics based algorithms for nonconvex optimization. In Advances in Neural Information Processing Systems, 2018.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. Cpo: A new approach for safe reinforcement learning with convergence guarantee. In International Conference on Machine Learning, pages 11480–11491. PMLR, 2021.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In International Conference on Machine Learning, pages 10746–10756. PMLR, 2020.

S. Zhang, B. Liu, and S. Whiteson. Mean-variance policy iteration for risk-averse reinforcement learning. In AAAI Conference on Artificial Intelligence (AAAI), 2021.
Shangtong Zhang, Bo Liu, and Shimon Whiteson. Mean-variance policy iteration for risk-averse reinforcement learning, 2020.

Han Zhong, Ethan X. Fang, Zhuoran Yang, and Zhaoran Wang. Risk-sensitive deep rl: Variance-constrained actor-critic provably finds globally optimal policy, 2020.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492, 2020.
Supplementary Material

A. Algorithm Details

We provide pseudo-code for the implementation of MVPI and VARAC in Algorithm 2 and 3.

Algorithm 2 MVPI

1: \textbf{Input}: number of iteration $K$, number of neural TD $T_{TD}$, learning rate for natural policy gradient (resp. PPO) and neural TD $\eta_{NPG}$ (resp. $\eta_{PPO}$) and $\eta_{TD}$, temperature parameters $\{\tau_k\}_{k=1}^K$.
2: \textbf{Initialization}: Initialize policy network $f((s, a); \theta, b)$ with $b_y \sim \text{Unif}(-1, 1)$ and $[\Theta_{\text{init}}]_{y, v} \sim \mathcal{N}(0, I_d/\delta), \forall v \in [m]$. Set $D = \{\xi \in \mathbb{R}^{md} : \|\xi - \Theta_{\text{init}}\|_2 \leq \Upsilon\}$, $\theta_1 = \Theta_{\text{init}}$, and $\tau_1 = 1$.
3: for $k = 1, \cdots, K$ do
4: Sample a batch of transitions $\{s_t, a_t, r_t, s_{t+1}\}^B_{t=1}$ following current policy with size of $B$.
5: $y = \frac{1}{B} \sum_{t=1}^B r_t$
6: for $t = 1, \cdots, B$ do
7: $\tilde{r}_t = r_t - \lambda r_{t-1}^2 + 2\lambda r_t y_t u_t \sim \pi(a|s_t)$
8: end for
9: while $\Delta \theta > \epsilon$ do
10: Q-value update: initialize the parameters of Q value function network $b$ and $\omega_{k,0}$ and update $\omega_k$ according to Eq. (3.2).
11: Output $Q_{\omega_k} = Q_{\omega_{k}(T_{TD})}$.
12: if select NPG update then
13: update $\theta_k$ according to Eq. (3.2).
14: else if select PPO update then
15: update $\theta_k$ according to Eq. (3.2).
16: end if
17: end while
18: end for
19: Output: $\pi_{\theta_K}$

Algorithm 3 VARAC

1: Input: number of iteration $K$, number of neural TD $T_{TD}$, learning rate for natural policy gradient (resp. PPO) and neural TD $\eta_{NPG}$ (resp. $\eta_{PPO}$) and $\eta_{TD}$, temperature parameters $\{\tau_k\}_{k=1}^K$.
2: Initialization: Initialize policy network $f((s, a); \theta, b)$ with $b_y \sim \text{Unif}(-1, 1)$ and $[\Theta_{\text{init}}]_{y, v} \sim \mathcal{N}(0, I_d/\delta), \forall v \in [m]$. Set $D = \{\xi \in \mathbb{R}^{md} : \|\xi - \Theta_{\text{init}}\|_2 \leq \Upsilon\}$, $\theta_1 = \Theta_{\text{init}}$, and $\tau_1 = 1$.
3: for $k = 1, \cdots, K$ do
4: Sample a batch of transitions $\{s_t, a_t, r_t, s_{t+1}\}^B_{t=1}$ following current policy with size of $B$.
5: $y = \frac{1}{B} \sum_{t=1}^B r_t$
6: Q-value update: initialize the parameters of both value function networks $Q$ and $W$ and update both networks’ $\omega_k$ according to Eq. (3.2).
7: Output $Q_k$ and $\Theta_k$.
8: update $\theta_k$ with NPG or PPO.
9: end for
10: Output: $\pi_{\theta_K}$

B. Experimental Details

Note that although the mean-volatility method can be adapted to off-policy methods (Zhang et al., 2020), in this paper, for the ease of the theoretical analysis, our proposed method is an on-policy actor-critic algorithm.

B.1. Testbeds

We use five Mujoco tasks from Open AI gym (Brockman et al., 2016) as testbeds. They are HalfCheetah-v2, Hopper-V2, Swimmer-V2, InvertedPendulum-v2, and InvertedDoublePendulum-v2.

B.2. Hyper-parameter Settings

In the experiment we set $\lambda = 1$. We then tune learning rate for different algorithms. For MVP, we use the same setting as Zhang et al. (2020). For MVPI, TOPS and VARAC with neural NPG, we tune the learning rate of the actor network from $\{0.1, 1 \times 10^{-2}, 1 \times 10^{-3}, 7 \times 10^{-4}\}$ and the learning rate of the critic network from $\{1 \times 10^{-2}, 1 \times 10^{-3}, 7 \times 10^{-4}\}$. For MVPI, TOPS and VARAC with neural PPO, we tune the learning rate of the actor network from $\{3 \times 10^{-3}, 3 \times 10^{-4}, 3 \times 10^{-5}\}$ and the learning rate of the critic network from $\{1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}\}$.

B.3. Computing infrastructure

We conducted our experiments on a GPU server where the CPU is “Intel(R) Xeon(R) Silver 4114 CPU” with 40 cores and 64 GB memory in total.

C. Theoretical Analysis Details

C.1. Additional Assumptions

Assumption 3. (Action-value function class) We define

$$\mathcal{F}_{\Upsilon, \infty} := \left\{ f(s, a; \theta) = f_0(s, a) + \int 1\{\theta^T(s, a) > 0\}(s, a)^T \iota(\theta) d\mu(w) : \|\iota(\theta)\|_{\infty} \leq \Upsilon/\sqrt{d} \right\}$$
It was proved (Wang et al., 2019; Liu et al., 2019) that the optimal policy w.r.t the risk-neutral objective $J(\pi)$ obtained by NPG and PPO method with the overparameterized two-layer neural network converges to the globally optimal policy at a rate of $O(1/\sqrt{K})$, where $K$ is the number of iteration. Since our method uses similar settings, we assume the convergence rates of risk-neutral objective $J(\pi)$ in our paper follow their results.

### C.2. Convergence analysis

In this section, we study the TOPS’s convergence of global optimality and provide a proof sketch. We first present the analysis of policy evaluation error, which is induced by TD update in Line 9 of Algorithm 1. We characterize the policy evaluation error in the following lemma:

**Lemma 2.** (Policy Evaluation Error) We set learning rate of $TD = \min\{(1 - \gamma)/3(1 + \gamma^2), 1/\sqrt{TD}\}$. Under Assumption 3 and 4, it holds that, with probability of $1 - \delta$,

$$
\|\hat{Q}_{n_k} - \hat{Q}_{n_k}\|_{\tilde{V}_{n_k}}^2 = O(\gamma^3 m^{-1/2} \log(1/\delta) + \gamma^{3/2} m^{-1/4} \sqrt{\log(1/\delta)} \\
+ \gamma_{\max}^2 m^{-1/4} + \gamma^2 T_{TD}^{1/2} + \gamma),
$$

where $\hat{Q}_{n_k}$ is the Q-value function of the augmented MDP, and $\hat{Q}_{n_k}$ is its estimator at the $k$-th iteration.

We first provide the supporting lemmas for Lemma 2. We define the local linearization of $f(x; \theta)$ defined in Eq. (2) at the initial point $\Theta_{\text{init}}$ as,

$$
\hat{f}(x; \theta) = \frac{1}{\sqrt{m}} \sum_{v=1}^{m} b_v \mathbb{1}\{[\Theta_{\text{init}}]_v (s, a) > 0\} \mathbb{1}[\theta]_v (s, a)
$$

We then define the following function spaces,

$$
\tilde{F}_{\infty} := \left\{ \frac{1}{\sqrt{m}} \sum_{v=1}^{m} b_v \mathbb{1}\{[\Theta_{\text{init}}]_v (s, a) > 0\} \mathbb{1}[\theta]_v (s, a) : ||\theta - \Theta_{\text{init}}||_2 \leq \Upsilon \right\},
$$

and

$$
\tilde{F}_{\infty} := \left\{ \frac{1}{\sqrt{m}} \sum_{v=1}^{m} b_v \mathbb{1}\{[\Theta_{\text{init}}]_v (s, a) > 0\} \mathbb{1}[\theta]_v (s, a) : |||\theta|_v - [\Theta_{\text{init}}]|_v||_\infty \leq \Upsilon / \sqrt{md} \right\}.
$$

$[\Theta_{\text{init}}]_v \sim \mathcal{N}(0, I_d/d)$ and $b_r \sim \text{Unif}([-1, 1])$ are the initial parameters. By the definition, $\tilde{F}_{\infty}$ is a subset of $\tilde{F}_{\infty}$. The following lemma characterizes the deviation of $\tilde{F}_{\infty}$ from $\tilde{F}_{\infty}$.
Lemma 3. (Projection Error) (Rahimi and Recht, 2009). Let \( f \in \mathcal{F}_{\top, \infty} \), where \( \mathcal{F}_{\top, \infty} \) is defined in Assumption 3. If \( \delta > 0 \), it holds with probability at least \( 1 - \delta \) that

\[
\| \Pi_{\mathcal{F}_{\top, \infty}} f - f \|_c \leq Y m^{-1/2} [1 + \sqrt{2 \log(1/\delta)}]
\]

where \( c \) is any distribution over \( S \times A \).

Lemma 4. (Linearization Error) Under Assumption 4, for all \( \theta \in \mathcal{D} \), where \( \mathcal{D} = \{ \xi \in \mathbb{R}^{m \times d} : \| \xi - \Theta_{\text{init}} \|_2 \leq \Upsilon \} \), it holds that,

\[
\mathbb{E}_{\nu_\theta} \left[ \| f((s, a); \theta) - \hat{f}((s, a); \theta) \|^2 \right] \leq \frac{4c_1 \Upsilon^3}{\sqrt{m}}
\]

where \( c_1 = c \sqrt{\mathbb{E}_{\mathcal{N}(0, I_d/d)}[1/\| (s, a) \|^2]} \), and \( c \) is defined in Assumption 4.

Proof. We start from the definitions in Eq. (2) and Eq. (C.2),

\[
\mathbb{E}_{\nu_\theta} \left[ \| f((s, a); \theta) - \hat{f}((s, a); \theta) \|^2 \right] = \mathbb{E}_{\nu_\theta} \left[ \left( \frac{1}{\sqrt{m}} \sum_{v=1}^{m} \left( \left\{ \| \Theta_{\text{init}} \|^2 (s, a) > 0 \right\} \theta_v[s, a] \right) \right)^2 \right]
\]

\[
\leq \frac{1}{m} \mathbb{E}_{\nu_\theta} \left[ \sum_{v=1}^{m} \left( \left\{ \| \Theta_{\text{init}} \|^2 (s, a) > 0 \right\} \theta_v[s, a] \right) \right] \leq \left\| \Theta_{\text{init}} \right\|_2
\]

First, since \( \left\{ \| \Theta_{\text{init}} \|^2 (s, a) > 0 \right\} \neq \left\{ \| \theta \|^2 (s, a) > 0 \right\} \), we have,

\[
\left\| \Theta_{\text{init}} \right\|_2 (s, a) \leq \| \theta \|_2 (s, a) - \Theta_{\text{init}} \|^2 (s, a) \leq \left\| \theta \right\|_2 - \Theta_{\text{init}} \|^2 _2,
\]

where we obtain the last inequality from the Cauchy-Schwarz inequality and the fact that \( \| (s, a) \|_2 \leq 1 \). Eq. (C.2) further implies that,

\[
\left\{ \| \theta \|^2 (s, a) > 0 \right\} - \left\{ \| \Theta_{\text{init}} \|^2 (s, a) > 0 \right\} \leq \left\{ \| \Theta_{\text{init}} \|^2 (s, a) \leq \| \theta \|_2 - \Theta_{\text{init}} \|^2 _2 \right\}
\]

Then plug Eq. (C.2) and the fact that \( |b_v| \leq 1 \) back to Eq. (C.2), we have the following,

\[
\mathbb{E}_{\nu_\theta} \left[ \| f((s, a); \theta) - \hat{f}((s, a); \theta) \|^2 \right] \leq \frac{1}{m} \mathbb{E}_{\nu_\theta} \left[ \sum_{v=1}^{m} \left\{ \| \Theta_{\text{init}} \|^2 (s, a) \leq \| \theta \|_2 - \Theta_{\text{init}} \|^2 _2 \right\} \right]
\]

We obtain the second inequality by the fact that \( |A| \leq |A - B| + |B| \). Then follow the the Cauchy-Schwarz inequality and the fact that \( \| (s, a) \|_2 \leq 1 \) we have the third equality. We continue Eq. (C.2) by following the Cauchy-Schwarz inequality and plugging \( \left\| \theta \right\|_2 - \Theta_{\text{init}} \|^2 _2 \leq \Upsilon \).

\[
\leq \frac{1}{m} \mathbb{E}_{\nu_\theta} \left[ \sum_{v=1}^{m} \left\{ \| \Theta_{\text{init}} \|^2 (s, a) \leq \| \theta \|_2 - \Theta_{\text{init}} \|^2 _2 \right\} \right]
\]

We obtain the second inequality by imposing Assumption 4 and the third by following the Cauchy-Schwarz inequality. Finally, we set \( c_1 := c \sqrt{\mathbb{E}_{\mathcal{N}(0, I_d/d)}[1/\| (s, a) \|^2]} \). Thus, we complete the proof. \( \Box \)

Recall \( T \) is the number of TD iteration. In the \( t \)-th iteration, we denote the temporal difference terms w.r.t \( \hat{f}((s, a); \theta_t) \),
and \( f((s, a); \theta_t) \) as

\[
\delta_0((s, a), (s, a)^{t}; \theta_t) = \hat{f}((s, a)^{t}; \theta_t) - \gamma \hat{f}((s, a); \theta_t) - r(s, a), (s, a)^{t}), \\
\delta_\theta((s, a), (s, a)^{t}; \theta_t) = f((s, a); \theta_t) - \gamma f((s, a); \theta_t) - r(s, a), (s, a)^{t}).
\]

For notation simplicity, in the sequel we write \( \delta_0((s, a), (s, a)^{t}; \theta_t) \) and \( \delta_\theta((s, a), (s, a)^{t}; \theta_t) \) as \( \delta_0 \) and \( \delta_\theta \). We further define the stochastic semi-gradient \( g_t(\theta_t) := \delta_\theta \nabla \delta f((s, a); \theta_t), \) its population mean \( \bar{g}_t(\theta_t) := E_{\nu_k}[g_t(\theta_t)], \) and the local linearization of \( \bar{g}_t(\theta_t), \bar{g}_0(\theta_t) := E_{\nu_k}[\delta_\theta \nabla \delta f((s, a); \theta_t)]. \) We denote them as \( g_t, \bar{g}_t, \bar{g}_0 \) respectively when there is no confusion.

**Lemma 5.** Under Assumption 4, for all \( \theta_t \in \mathcal{D}, \) where \( \mathcal{D} = \{x \in \mathbb{R}^m : \|x - \theta_{\text{min}}\|_2 \leq \Upsilon \}, \) it holds with probability of \( 1 - \delta \) that,

\[
\|\bar{g}_t - \bar{g}_0\|_2^2 = O\left(\Upsilon^3/2m^{-1/4}(1 + (m \log 1/\delta)^{-1/2}) + \Upsilon^{1/2}r_{\text{max}}m^{-1/4}\right).
\]

**Proof.** By the definition of \( \bar{g}_t \) and \( \bar{g}_0, \) we have

\[
\|\bar{g}_t - \bar{g}_0\|_2^2 = \|E_{\nu_k}\delta_\theta \nabla \delta f((s, a); \theta_t) - \delta_\theta \nabla \delta f((s, a); \theta_t)|\|_2^2 \\
= \|E_{\nu_k}[\delta_\theta - \delta_0]\nabla \delta f(x; \theta_t) + \delta_\theta (\nabla \delta f(x; \theta_t) - \nabla \delta \hat{f}(x; \theta_t))\|_2^2 \\
\leq \left(E_{\nu_k}[\|\delta_\theta - \delta_0\|_2 \|\nabla \delta f(x; \theta_t)\|_2 + \|\delta_\theta\| \|\nabla \delta f(x; \theta_t) - \nabla \delta \hat{f}(x; \theta_t)\|_2]\right)^2 \\
\leq 2E_{\nu_k}[\|\delta_\theta - \delta_0\|^2 \|\nabla \delta f(x; \theta_t)\|_2^2] + 2E_{\nu_k}[\|\delta_\theta\| \|\nabla \delta f(x; \theta_t) - \nabla \delta \hat{f}(x; \theta_t)\|_2^2].
\]

We first upper bound \( E_{\nu_k}[\|\delta_\theta - \delta_0\|^2 \|\nabla \delta f(x; \theta_t)\|_2^2] \) in Eq. (C.2). Since \( \|x\|_2 \leq 1, \) we have \( \|\nabla \delta f(x; \theta_t)\|_2 \leq 1. \) Then by definition,

\[
E_{\nu_k}[\|\delta_\theta - \delta_0\|^2 \|\nabla \delta f(x; \theta_t)\|_2^2] \\
\leq E_{\nu_k}[\|f(x; \theta_t) - \hat{f}(x; \theta_t) - \gamma f((s', a'); \theta_t) - \hat{f}((s', a'); \theta_t)|^2] \\
\leq E_{\nu_k}[\|f(x; \theta_t) - \hat{f}(x; \theta_t) + |f((s', a'); \theta_t) - \hat{f}((s', a'); \theta_t)|^2] \\
\leq 2E_{\nu_k}[\|f(x; \theta_t) - \hat{f}(x; \theta_t)|^2] + 2E_{\nu_k}[\|f((s', a'); \theta_t) - \hat{f}((s', a'); \theta_t)|^2] \\
\leq 4E_{\nu_k}[\|f(x; \theta_t) - \hat{f}(x; \theta_t)|^2] \leq \frac{16c_1 \Upsilon^3}{\sqrt{m}}.
\]

We obtain the first inequality by \( |\gamma| \leq 1, \) and reach the final step by inserting Lemma 4. We then proceed to upper bound \( E_{\nu_k}[\|\delta_0\| \|\nabla \delta f(x; \theta_t) - \nabla \delta \hat{f}(x; \theta_t)\|_2] \). From Hölder’s inequality, we have,

\[
E_{\nu_k}[\|\delta_0\| \|\nabla \delta f(x; \theta_t) - \nabla \delta \hat{f}(x; \theta_t)\|_2] \leq E_{\nu_k}[\|\delta_0\|^2 E_{\nu_k}[\|\nabla \delta f(x; \theta_t) - \nabla \delta \hat{f}(x; \theta_t)\|_2^2]
\]

We first derive an upper bound for the first term in Eq. (C.2), starting from its definition,

\[
E_{\nu_k}[\|\delta_0\|^2] = E_{\nu_k}[\|\hat{f}((s', a'); \theta_t) - \gamma \hat{f}(x; \theta_t) - r(x, (s', a'))\|^2] \\
\leq 3\left(E_{\nu_k}[\|\hat{f}((s', a'); \theta_t)\|^2] + E_{\nu_k}[\|\gamma \hat{f}(x; \theta_t)\|^2] + E_{\nu_k}[\|r^2(x, (s', a'))\|^2]\right) \\
\leq 6E_{\nu_k}[\|\hat{f}(x; \theta_t)\|^2] + 3M^2 \\
= E_{\nu_k}[\|\hat{f}(x; \theta_t) - \hat{f}(x; \theta_{\text{argmax}}) + \hat{f}(x; \theta_{\text{argmax}}) - Q_\pi \|^2] + 3M^2 \\
\leq 72\Upsilon^2 + 18E_{\nu_k}[\|\hat{f}(x; \Theta_\pi^*) - Q_\pi\|^2] + 21(1 - \gamma)^{-2}r_{\text{max}}^2.
\]

Recall \( r_{\text{max}} \) is the boundary for reward function \( r. \) We obtain the last inequality in Eq. (C.2) following the fact that \( |\hat{f}(x; \theta_t) - \hat{f}(x; \theta_{\text{argmax}})| \leq \|\theta_t - \theta_{\text{argmax}}\| \leq \Upsilon \) and \( Q_\pi \leq (1 - \gamma)^{-1}r_{\text{max}}. \) Since \( F_{\Upsilon, m} \subset F_{\Upsilon, m}, \) by Lemma 3, we have,

\[
E_{\nu_k}[\|\hat{f}(x; \Theta_\pi^*) - Q_\pi\|^2] \leq \Upsilon^2 \left(1 + \sqrt{2\log(1/\delta)} \right)^2
\]

Combine Eq. (C.2) and Eq. (C.2), assuming \( 0 < \delta < 1/\sqrt{1 - \sqrt{2\log(1/\delta)}}, \) we have, with probability of \( 1 - \delta, \)

\[
E_{\nu_k}[\|\delta_0\|^2] \leq 72\Upsilon^2 \left(1 + \sqrt{\frac{\log(1/\delta)}{m}} \right) + 21(1 - \gamma)^{-2}r_{\text{max}}^2.
\]
Lastly we have
\[
\mathbb{E}_{\nu_t} \left[ \left\| \nabla_{\theta} f(x; \theta_t) - \nabla_{\theta} \tilde{f}(x; \theta_t) \right\|_2^2 \right]
= \mathbb{E}_{\nu_t} \left[ \left( \frac{1}{m} \sum_{v=1}^{m} \left( \mathbb{I}[\{[\theta_t]^T x > 0 \} - \mathbb{I}[\{[\Theta_{\text{init}}]^T x > 0 \}] \right)^2 + \|b_v\|^2 \|x\|^2_2 \right) \right]
\leq \mathbb{E}_{\nu_t} \left[ \frac{1}{m} \sum_{v=1}^{m} \left( \mathbb{I}[\{[\Theta_{\text{init}}]^T x \leq \|\Theta_v - [\Theta_{\text{init}}]\|_2 \}] \right) \right]
\leq \frac{c_1 \gamma}{\sqrt{m}}
\]

We obtain the first inequality by following Eq. (C.2) and the fact that \(|b_v| \leq 1\) and \(|\mathbb{E}_v| \leq 1\). Then for the rest, we follow the similar argument in Eq. (C.2). To finish the proof, we plug Eq. (C.2), Eq. (C.2) and Eq. (C.2) back to Eq. (C.2).

\[
\|g_t - \tilde{g}_t\|_2^2 \leq 2 \left( \frac{16c_1 \gamma^3}{\sqrt{m}} + \left( 72\gamma^2 \left( 1 + \log \left( \frac{1}{\delta} \right) \right) + 21(1 - \gamma)^{-2} \gamma^2 \right) \right)
\leq \frac{c_1 \gamma}{\sqrt{m}}
\]

Then we have,

\[
\|\tilde{g}_t - \tilde{g}_0\|_2 \leq \sqrt{\frac{16c_1 \gamma^3}{\sqrt{m}} + \left( 144c_1 \gamma^3 \log \left( \frac{1}{\delta} \right) \right) m^{3/2} + \frac{42c_1 \gamma_r^2}{(1 - \gamma)^{-2} \sqrt{m}}}
\leq \sqrt{16c_1 \gamma^3 + \left( 144c_1 \gamma^3 \log \left( \frac{1}{\delta} \right) \right) m^{3/2} + \frac{42c_1 \gamma_r^2}{(1 - \gamma)^{-2} \sqrt{m}}}
= \mathcal{O} \left( \frac{\gamma^{3/2} m^{-1/4}}{\sqrt{m}} + 1 + \left( m \log \frac{1}{\delta}\right)^{-1/2} \right)
\]

Proof.

\[
\|g_t - \bar{g}_t\|_2 \leq \sqrt{\frac{16c_1 \gamma^3}{\sqrt{m}} + \left( 144c_1 \gamma^3 \log \left( \frac{1}{\delta} \right) \right) m^{3/2} + \frac{42c_1 \gamma_r^2}{(1 - \gamma)^{-2} \sqrt{m}}}
\]

Two of the terms on the right hand side of Eq. (C.2) are characterized in Lemma 5 and Lemma 6. We therefore characterize the remaining term,

\[
\|g_t - \bar{g}_t\|_2 \leq \mathbb{E}_{\nu_t} \left[ (\bar{g}_t - \bar{g}_0)^2 \right] \leq \mathbb{E}_{\nu_t} \left[ \left( \bar{g}_t - \bar{g}_0 \right)^2 \right] + \mathbb{E}_{\nu_t} \left[ \left( \bar{g}_t - \bar{g}_0 \right)^2 \right] = \mathbb{E}_{\nu_t} \left[ \left( \bar{g}_t - \bar{g}_0 \right)^2 \right] + \mathbb{E}_{\nu_t} \left[ \left( \bar{g}_t - \bar{g}_0 \right)^2 \right]
\]

Next, we provide the following lemma to characterize the variance of \(g_t\).

**Lemma 6.** (Variance of the Stochastic Update Vector)(Liu et al., 2019). There exists a constant \(\xi^2 = \mathcal{O}(\gamma^2)\) independent of \(t\) such that for any \(t \leq T\), it holds that

\[
\mathbb{E}_{\nu_t} \|g_t - \bar{g}_t\|_2^2 \leq \xi^2
\]

Now we provide the proof for Lemma 2.
(s’, a’) have the same marginal distribution, as well as γ < 1 for the second inequality. Next, we upper bound
\[ g_t(\theta_t) - \bar{g}_0(\theta_{\pi^*}), \theta_t - \theta_{\pi^*} \]. We have,
\[ \langle g_t(\theta_t) - \bar{g}_0(\theta_{\pi^*}), \theta_t - \theta_{\pi^*} \rangle = \langle g_t(\theta_t) - \bar{g}_0(\theta_{\pi^*}), \theta_t - \theta_{\pi^*} \rangle + \langle \bar{g}_0(\theta_t) - \bar{g}_0(\theta_{\pi^*}), \theta_t - \theta_{\pi^*} \rangle \]

One term on the right hand side of Eq. (C.2) are characterized by Lemma 6. We continue to characterize the remaining terms. First, by Hölder’s inequality, we have
\[ \langle \bar{g}_0(\theta_t) - \bar{g}_0(\theta_{\pi^*}), \theta_t - \theta_{\pi^*} \rangle \geq -\|\bar{g}_0(\theta_t) - \bar{g}_0(\theta_{\pi^*})\|_2 \|\theta_t - \theta_{\pi^*}\|_2 \geq -2\mathcal{Y}\|\bar{g}_0(\theta_t) - \bar{g}_0(\theta_{\pi^*})\|_2 \]

We obtain the second inequality since \( \|\theta_t - \theta_{\pi^*}\|_2 \leq 2\mathcal{Y} \) by definition. For the last term,
\[ \langle \bar{g}_0(\theta_t) - \bar{g}_0(\theta_{\pi^*}), \theta_t - \theta_{\pi^*} \rangle = \mathbb{E}_{\nu_\mathcal{Y}} \left[ c(x, \theta_t; t) - \hat{c}(x, \theta_{\pi^*}) \right] \]

where the inequality follows from Eq. (C.2). Combine Eqs. (C.2), (C.2), (C.2), (C.2) and (C.2), we have,
\[ \|\Theta_t + 1 - \Theta_{\pi^*}\|_2 \leq \|\theta_t - \theta_{\pi^*}\|_2 - (2\eta(1 - \gamma) - 3\eta^2(1 + \gamma)^2)\mathbb{E}_{\nu_\mathcal{Y}} \left[ (\hat{f}(x, \theta_t) - \hat{f}(x, \theta_{\pi^*}))^2 \right] \]

We then bound the error terms by rearrange Eq. (C.2). First, we have, with probability of \( 1 - \delta \),
\[ \mathbb{E}_{\nu_\mathcal{Y}} \left[ (f(x, \theta_t) - \hat{f}(x, \theta_{\pi^*}))^2 \right] \leq \mathbb{E}_{\nu_\mathcal{Y}} \left[ (f(x, \theta_t) - \hat{f}(x, \theta_{\pi^*}))^2 + (\hat{f}(x, \theta_t) - \hat{f}(x, \theta_{\pi^*}))^2 \right] \leq \delta \]
\[ \mathbb{E}_{\nu_\mathcal{Y}} \left[ (f(x, \theta_t) - \hat{f}(x, \theta_{\pi^*}))^2 + (\hat{f}(x, \theta_t) - \hat{f}(x, \theta_{\pi^*}))^2 \right] \]

where \( \epsilon_\mathcal{Y} = \mathcal{O}(T^3m^{1/2}\log(1/\delta) + T^5/2m^{-1/4}\sqrt{\log(1/\delta)}) \)

By telescoping Eq. (C.2) for \( t = 0 \) to \( T - 1 \), we have, with probability of \( 1 - \delta \),
\[ \|f(x, \theta) - \hat{f}(x, \theta_{\pi^*})\| \leq \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\nu_\mathcal{Y}} \left[ (f(x, \theta) - \hat{f}(x, \theta_{\pi^*}))^2 \right] \]

Set \( \eta = \min\{1/\sqrt{T}; (1 - \gamma)/3(1 + \gamma)^2\} \), which implies that \( T^{-1/2}(2\eta^2(1 - \gamma) - 3\eta^2(1 + \gamma)^2)^{-1} \leq 1/(1 - \gamma)^2 \), then we have, with probability of \( 1 - \delta \),
\[ \|f(x, \theta) - \hat{f}(x, \theta_{\pi^*})\| \leq \frac{1}{(1 - \gamma)^2\sqrt{T}} \left( \|\Theta_{\pi^*} - \Theta_{\pi^*}\|_2 + 4\mathcal{Y}\sqrt{T}\|\xi_\mathcal{Y}\| \right) + 3\mathcal{Y}\|\xi_\mathcal{Y}\| + \epsilon_\mathcal{Y} \]

We obtain the second inequality by the fact that \( \|\Theta_{\pi^*} - \Theta_{\pi^*}\|_2 \leq \mathcal{Y} \). Then by definition we replace \( \hat{Q}_{\omega_k} \) and \( \hat{Q}_{\pi_k} \) in the following, we establish the error induced by the policy update. This work presents proofs for two gradient-based methods, NPG and PPO, with the overparameterized two-layer neural network. We first denote the objective of the risk-neutral MDP with the augmented reward defined in Eq. (3.1) with respect to auxiliary variable \( y \) by,
\[ J^y_\mathcal{X}(\pi) = \sum_{s, a} \sigma_\pi(r - \lambda r - 2\lambda y_{k+1}) - \lambda y_{k+1}^2 \]
We then follow Lemma 6.1 in Kakade and Langford (2002):\
\[c\]
where \(\sigma_v\) is the state-action value function of the augmented MDP, and its rewards are associated with \(y\).

\[\text{Proof.}\] When \(y\) is fixed,
\[
J^+_\lambda(\pi') - J^+_\lambda(\pi) = \sum_{s,a} \sigma_{v'}(s,a) - \sum_{s,a} \sigma_{v'}(s,a) = J(\pi') - J(\pi)
\]
We then follow Lemma 6.1 in Kakade and Langford (2002):
\[
\hat{J}(\pi') - \hat{J}(\pi) = (1 - \gamma)^{-1} \mathbb{E}_{s \sim \nu', \pi}[\hat{Q}_{\pi,y}] - \mathbb{E}_{s \sim \pi}[\hat{Q}_{\pi,y}]
\]
where \(\hat{A}_{\pi} = \hat{Q}_{\pi} - \hat{V}_{\pi}\) is the advantage function of policy \(\pi\). Meanwhile,
\[
\mathbb{E}_{s \sim \pi}[\hat{A}_{\pi}] = \mathbb{E}_{s \sim \pi}[\hat{Q}_{\pi}] - \mathbb{E}_{s \sim \pi}[\hat{V}_{\pi}]
\]
From Eq. (C.2), Eq. (C.2) and Eq. (C.2), we complete the proof. \(\square\)

Lemma 7 is inspired by Kakade and Langford (2002) and adopted by most work on global convergence (Agarwal et al., 2020; Liu et al., 2019; Xu et al., 2021). Next, we derive an upper bound for the error of the critic update in Line 5 of Algorithm 1:

Lemma 8. (Update Error) We characterize the error induced by the estimation of auxiliary variable \(y\) w.r.t. the optimal value \(y^*\) at \(k\)-th iteration as,
\[
J^+_\lambda(\pi^*) - J^{y_k}_\lambda(\pi^*) = \frac{2c_0r_{\max}^y(1 - \gamma)^3\lambda}{\sqrt{k}},
\]
where \(r_{\max}^y\) is the bound of the original reward, and \(c_3\) is a constant error term.

\[\text{Proof.}\] We start from the subproblem objective defined in Eq. (C.2) with \(y^*\) and \(\hat{y}_k^*\):
\[
J^+_\lambda(\pi^*) - J^{y_k}_\lambda(\pi^*) = \sum_{s,a} \sigma_v(r(s,a) - \lambda r(s,a)^2 + 2\lambda r(s,a)y^*) - \lambda y^* - \lambda y^2
\]
Thus we finish the proof. \(\square\)

Neural NPG

In the following part, we focus on the convergence of neural NPG. We first define the following terms under neural NPG:

\[\text{Proof.}\] In the following part, we focus the convergence of neural NPG. We first define the following terms under neural NPG:
\[
\nabla_{\theta}J(\pi_\theta) = \tau \mathbb{E}_{d_{\theta}(s,a)}[\hat{Q}_{\pi_\theta}(s,a)(\phi_\phi(s,a) - \mathbb{E}_{\pi_\theta}[\phi_\phi(s,a)')])
\]
\[F(\theta) = \tau^2 \mathbb{E}_{d_{\theta}(s,a)}[(\phi_\phi(s,a) - \mathbb{E}_{\pi_\theta}[\phi_\phi(s,a)])^T (\phi_\phi(s,a) - \mathbb{E}_{\pi_\theta}[\phi_\phi(s,a)])]
\]
We then derive an upper bound for \(J^+_\lambda(\pi^*) - J^{y_k}_\lambda(\pi_k)\) for the neural NPG method in the following lemma:

Lemma 10. (One-step difference of \(\pi\)) It holds that, with probability of \(1 - \delta\),
\[
(1 - \gamma)(J^+_\lambda(\pi^*) - J^{y_k}_\lambda(\pi_k)) \\
\leq \eta_{\text{NPG}}^2 \mathbb{E}_{\pi \sim \nu_{\pi_k}}[\text{KL}(\pi^* \| \pi_k) - \text{KL}(\pi^* \| \pi_{k+1})] + \\
\eta_{\text{NPG}}(9\sqrt{2}M^2 + 2c_0\epsilon_{Q,k} + \eta_{\text{NPG}}^2\epsilon_k)
\]
\[c_{Q,k} = \mathcal{O}(\Upsilon^3 m^{-1/2} \frac{\log(1/\delta)}{m} + \frac{\Upsilon^5}{m^{1/2}} \sqrt{\log(1/\delta)} + \\
\Upsilon_{r_{\max}^y}^2 m^{-1/4} + \Upsilon^2 T_D^{1/2} + \Upsilon)\text{ and }\epsilon_k = \\
\sqrt{8\eta_{\text{NPG}} \Upsilon^3 m^{-1/4} + \eta_{\text{NPG}}^2 \Upsilon^5 m^{-1/8}}\text{, where }c_0\text{ is the upper bound for concentrability coefficients (Munos, 2007) and }\epsilon_k\text{ is the upper bound for gradient variance.}
\]
Meanwhile, \(\Upsilon\) is the radius of the parameter space, \(m\) is the width of the neural network, and \(B\) is the sample batch size.
Proof. We start from the following,

\[
KL(\pi^* \| \pi_k) - KL(\pi^* \| \pi_{k+1}) - KL(\pi_{k+1} \| \pi_k) \\
= E_{a \sim \pi^*} \left[ \log(\frac{\pi_{k+1}}{\pi_k}) \right] - E_{a \sim \pi_{k+1}} \left[ \log(\frac{\pi_k}{\pi_{k+1}}) \right]
\]

(by KL's definition).

We then add and subtract a few terms. Rearrange these terms, we get,

\[
\begin{align*}
&= \left( E_{a \sim \pi^*} \left[ \log(\frac{\pi_{k+1}}{\pi_k}) \right] - \eta_{NPG} \hat{Q}_{\omega_k} \right) \\
&- E_{a \sim \pi_{k+1}} \left[ \log(\frac{\pi_k}{\pi_{k+1}}) \right] + \eta_{NPG} \left( E_{a \sim \pi^*} \left[ \hat{Q}_{\omega_k} - \hat{Q}_{\pi_k, \tilde{y}_k} \right] - E_{a \sim \pi_{k+1}} \left[ \hat{Q}_{\omega_k} - \hat{Q}_{\pi_k, \tilde{y}_k} \right] \right)
\end{align*}
\]

We now show the building blocks of the proof. First, we take the expectation of both sides of Eq. (C.2) with respect to \( s \sim \nu_{\pi^*} \). By Lemma 7, we have

\[
\eta_{NPG} E_{s \sim \nu_{\pi^*}} \left[ E_{a \sim \pi^*} \left[ \hat{Q}_{\omega_k} - \hat{Q}_{\pi_k, \tilde{y}_k} \right] - E_{a \sim \pi_{k+1}} \left[ \hat{Q}_{\omega_k} - \hat{Q}_{\pi_k, \tilde{y}_k} \right] \right] = \eta_{NPG} (1 - \gamma) (J^*_\lambda (\pi^*) - J^*_\lambda (\pi_{k+1}))
\]

We then group up all the terms on RHS of Eq. (C.2) except LHS of Eq. (C.2) denote by \( H_k \),

\[
H_k := E_{s \sim \nu_{\pi^*}} \left[ E_{a \sim \pi^*} \left[ \log(\frac{\pi_{k+1}}{\pi_k}) \right] - \eta_{NPG} \hat{Q}_{\omega_k} \right] \\
- E_{a \sim \pi_{k+1}} \left[ \log(\frac{\pi_k}{\pi_{k+1}}) \right] + \eta_{NPG} E_{s \sim \nu_{\pi^*}} \left[ E_{a \sim \pi^*} \left[ \hat{Q}_{\omega_k} - \hat{Q}_{\pi_k, \tilde{y}_k} \right] - E_{a \sim \pi_{k+1}} \left[ \hat{Q}_{\omega_k} - \hat{Q}_{\pi_k, \tilde{y}_k} \right] \right]
\]

Insert Eqs. (C.2) and (C.2) back to Eq. (C.2), we have,

\[
\eta_{NPG} (1 - \gamma) (J^*_\lambda (\pi^*) - J^*_\lambda (\pi_{k+1})) = E_{s \sim \nu_{\pi^*}} \left[ KL(\pi^* \| \pi_k) - KL(\pi^* \| \pi_{k+1}) - KL(\pi_{k+1} \| \pi_k) \right] \\
- H_k
\]

We reach the final inequality of Eq. (C.2) by algebraic manipulation. Second, we follow Lemma 5.5 of Wang et al. (2019) and obtain an upper bound for Eq. (C.2). Specifically, with probability of 1 \(-\delta\),

\[
E_{a \sim \text{init}} \left[ H_k \right] - E_{s \sim \nu_{\pi^*}} \left[ KL(\pi_{k+1} \| \pi_k) \right] \leq \eta_{NPG}^2 (9\Upsilon^2 + M^2) + 2\eta_{NPG} c_0 \epsilon_{Q,k} + \epsilon_k
\]

The expectation is taken over randomness. With these building blocks of Eqs. (C.2) and (C.2), we are now ready to reach the concluding inequality. Plugging Eqs. (C.2) back into Eq. (C.2), we end up with, with probability of 1 \(-\delta\),

\[
\eta_{NPG} (1 - \gamma) (J^*_\lambda (\pi^*) - J^*_\lambda (\pi_{k+1})) \leq E_{s \sim \nu_{\pi^*}} \left[ KL(\pi^* \| \pi_k) - KL(\pi^* \| \pi_{k+1}) \right] + \eta_{NPG}^2 (9\Upsilon^2 + M^2) + 2\eta_{NPG} c_0 \epsilon_{Q,k} + \epsilon_k
\]

Dividing both sides of Eq. (C.2) by \( \eta_{NPG} \) completes the proof. The details are included in the Appendix.

We have the following Lemma to bound the error terms \( H_k \) defined in Eq. (C.2) of Lemma 10.

**Lemma 11.** (Wang et al., 2019). Under Assumptions 4, we have

\[
E_{a \sim \text{init}} \left[ H_k \right] - E_{s \sim \nu_{\pi^*}} \left[ KL(\pi_{k+1} \| \pi_k) \right] \leq \eta_{NPG}^2 (9\Upsilon^2 + M^2) + \eta_{NPG} (\varphi_k' + \psi_k') \epsilon_{Q,k} + \epsilon_k
\]

Here the expectation is taken over all the randomness. We have \( \epsilon_{Q,k} := \| Q_{\omega_k} - Q_{\pi_k} \|^2 \| \nu_{\pi_k} \| \)

\[
\epsilon_k = \sqrt{2\Upsilon^2} \eta_{NPG} (\varphi_k + \psi_k)^{1/2} \{ E_{s,a \sim \pi_{\pi_k}} \left[ ||\xi_k(\delta_k)\|^2 \right] \\
+ E_{s,a \sim \pi_{\pi_k}} \left[ ||\xi_k(\omega_k)\|^2 \right] \}^{1/2} \\
+ \mathcal{O}( (\tau_{k+1} + \eta_{NPG}) \Upsilon^{3/2} m^{-1/4} + \eta_{NPG} \Upsilon^{5/4} m^{-1/8} )
\]

Recall \( \xi_k(\omega_k) \) and \( \xi_k(\omega_k) \) are defined in Assumption 1, while \( \varphi_k, \psi_k, \varphi_k', \) and \( \psi_k' \) defined in Assumption 2.

Please refer to Wang et al. (2019) for complete proof.

Finally, we are ready to show the proof for Theorem 1.

**Proof.** First, we combine Lemma 8 and 10 to get the following:

\[
(1 - \gamma) (J^*_\lambda (\pi^*) - J^*_\lambda (\pi_{k+1})) \leq \eta_{NPG}^2 \left[ KL(\pi^* \| \pi_k) - KL(\pi^* \| \pi_{k+1}) \right] + \eta_{NPG} (9\Upsilon^2 + M^2) + 2\eta_{NPG} c_0 \epsilon_{Q,k} + \eta_{NPG} \epsilon_k
\]

We can then see this:

1. \( E_{s \sim \nu_{\pi^*}} [KL(\pi^* \| \pi_1)] \leq \log \| A \| \) due to the uniform initialization of policy.
2. KL($\pi^* \parallel \pi_{K+1}$) is a non-negative term.

And by setting $\eta_{NPG} = 1/\sqrt{K}$ and telescoping Eq. (C.2), we obtain,

$$(1 - \gamma) \min_{k \in [K]} \left( J^\nu_\lambda(\pi^*) - J^\nu_\lambda(\pi_k) \right)$$

$$\leq (1 - \gamma) \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}(J^\nu_\lambda(\pi^*) - J^\nu_\lambda(\pi_k))$$

$$\leq \frac{1}{\sqrt{K}} \left( (\mathbb{E}_{s \sim \nu_{\pi_s}}[KL(\pi^* \parallel \pi_1)] + 9 \gamma^2 + M^2) + \frac{1}{K} \sum_{k=1}^{K} (2\sqrt{K} c_0 \epsilon_{Q,k} + \eta^{-1}_{NPG} \epsilon_k + 2\epsilon_k M(1 - \gamma)^2 \lambda_j) \right)$$

By plugging $\epsilon_{Q,k}$ and $\epsilon_k$, defined in Lemma 10 into Eq. (C.2), then setting $\epsilon_k(K)$ as the error term, we complete the proof. □

**NEURAL PPO**

We now study the global convergence of TOPS with neural PPO as the policy update component. First, we define the neural PPO update rule.

**Lemma 12.** (Liu et al., 2019). Let $\pi_{\theta_k} \propto \exp\{\tau_k^{-1} f_{\theta_k}\}$ be an energy-based policy. We define the update

$$\hat{\pi}_{k+1} = \arg \max_{\pi} \mathbb{E}_{\pi \sim \nu_{\pi}} [\mathbb{E}_{\pi}[Q_{\omega_k}] - \beta_k KL(\pi_{\theta_k} \parallel \pi_{\theta_k})]$$

, where $Q_{\omega_k}$ is the estimator of the exact action-value function $Q^{\pi_{\theta_k}}$. We have

$$\hat{\pi}_{k+1} \propto \exp\{\beta_k^{-1} Q_{\omega_k} + \tau_k^{-1} f_{\theta_k}\}$$

And to represent $\hat{\pi}_{k+1}$ with $\pi_{\theta_{k+1}} \propto \exp\{\tau_{k+1}^{-1} f_{\theta_{k+1}}\}$, we solve the following subproblem,

$$\theta_{k+1} = \arg \min_{\theta \in \mathcal{D}} \mathbb{E}_{(s,a) \sim \pi_{\theta}} \left[ (f_{\theta}(s,a) - \tau_{k+1}^{-1}(\beta_k^{-1} Q_{\omega_k}(s,a) + \tau_k^{-1} f_{\theta_k}(s,a)))^2 \right]$$

We analyze the error in Line 14 of Algorithm 1. Liu et al. (2019) proves that the policy improvement error can be characterized similarly to the policy evaluation error as in Eq. (2). Recall $Q_{\omega_k}$ is the estimator of Q-value, $f_{\theta_k}$ the energy function for policy, and $f_{\tilde{\theta}}$ its estimator. We characterize the policy improvement error as follows: Under Assumptions 3 and 4, we set the learning rate of PPO $\eta_{PPO} = \min\{(1 - \gamma)/3(1 + \gamma)^2/\sqrt{TT_{TD}}\}$, and with a probability of $1 - \delta$:

$$\| (f_{\tilde{\theta}} - \tau_{k+1}(\beta_k^{-1} Q_{\omega_k} + \tau_k^{-1} f_{\theta_k})) \|^2$$

$$= O(\gamma^3 m^{-1/2} \log(1/\delta) + \gamma^{3/2} m^{-1/4} \sqrt{\log(1/\delta)}) + \gamma^2 m^{-1/4} + \gamma^2 T_{TD}^{-1/2} + \gamma).$$

We quantify how the errors propagate in neural PPO (Liu et al., 2019) in the following.

**Lemma 13.** (Liu et al., 2019). (Error Propagation) We have the policy improvement error satisfy By bounding the policy improvement error as

$$\mathbb{E}_{(s,a) \sim \pi_{\theta_k}} \| (f_{\tilde{\theta}}(s,a) - \tau_{k+1}(\beta_k^{-1} Q_{\omega_k}(s,a) + \tau_k^{-1} f_{\theta_k}(s,a)))^2 \| \leq \epsilon_{k+1}$$

and bounding the policy evaluation error as

$$\mathbb{E}_{(s,a) \sim \pi_{\theta_k}} \| (\hat{Q}_{\omega_k} - \hat{Q}_{\omega_k})^2 \| \leq \epsilon'_k.$$

Then we have,

$$\mathbb{E}_{s \sim \nu_{\pi_k}} \left[ \mathbb{E}_{a \sim \pi} \left[ \log(\pi_{\theta_{k+1}}/\pi_{k+1}) - \mathbb{E}_{a \sim \pi_k} \log(\pi_{\theta_{k+1}}/\pi_k) \right] \right] \leq \epsilon_{k+1} \epsilon_k \epsilon_k + \beta^{-1} \epsilon_k \epsilon'_k$$

$\epsilon_{k+1}$ and $\epsilon'_k$ are defined in Eq.(2) and (C.2), and $\varphi_k^*$ =

$$\mathbb{E}_{(s,a) \sim \pi_k} \left[ \left( \frac{d\pi^*}{d\pi_0} - \frac{d\pi_{\theta_k}}{d\pi_0} \right)^2 \right]^{1/2}, \psi_k = \mathbb{E}_{(s,a) \sim \pi_k} \left[ \left( \frac{d\pi^*}{d\pi_0} - \frac{d\pi_{\theta_k}}{d\pi_0} \right)^2 \right]^{1/2}.$$

The Radon-Nikodym derivatives (Konstantopoulos et al., 2011). We denote RHS in Eq. (13) by $\xi_k = \epsilon_{k+1} \epsilon_k \epsilon_k + \beta^{-1} \epsilon_k \epsilon'_k$. Lemma 13 essentially quantifies the error from which we use the two-layer neural network to approximate the action-value function and policy instead of having access to the exact ones. Please refer to Liu et al. (2019) for complete proofs of Lemma 12 and 13.

$$\mathbb{E}_{s \sim \nu_{\pi_k}} \left[ \mathbb{E}_{a \sim \pi} \left[ \log(\pi_{\theta_{k+1}}/\pi_{k+1}) \right] \right] \leq \epsilon_{k+1} \epsilon_k \epsilon_k + \beta^{-1} \epsilon_k \epsilon'_k$$

We then characterize the difference between energy functions in each step (Liu et al., 2019). Under the optimal policy $\pi^*$,

**Lemma 14.** (Liu et al., 2019). (Stepwise Energy Function difference) Under the same condition of Lemma 13, we have

$$\mathbb{E}_{s \sim \nu_{\pi^*}} \| \tau_{k+1}^{-1} f_{\theta_{k+1}} - \tau_k^{-1} f_{\theta_k} \|^2 \leq 2\epsilon'_k + 2\beta^{-2} U,$$

where $\epsilon'_k = |A| \tau_{k+1}^{-2} \epsilon_k^2$ and $U = 2\mathbb{E}_{s \sim \nu_{\pi^*}} \left( \max_{a \in A}(Q_{\omega_k})^2 \right) + 2\gamma^2.$

**Proof.** By the triangle inequality, we get the following,

$$\| \tau_{k+1}^{-1} f_{\theta_{k+1}} - \tau_k^{-1} f_{\theta_k} \|^2 \leq 2(\| \tau_{k+1}^{-1} f_{\theta_{k+1}} - \tau_k^{-1} f_{\theta_k} - \beta^{-1} Q_{\omega_k} \|^2 + \| \beta^{-1} Q_{\omega_k} \|^2).$$
We take the expectation of both sides of Eq. (C.2) with respect to \( s \sim P^\pi \). With the 1-Lipshitz continuity of \( \tilde{Q}_{k+1} \) in \( \theta_k \) and \( \|\theta_k - \Theta_{\text{init}}\|_2 \leq \Upsilon \), we have,

\[
\mathbb{E}_{\nu^\pi} \left[ \left\| \tau_k^{-1} f_{\theta_{k+1}} - \tau_k^{-1} f_{\theta_k} \right\|_2^2 \right] \\
\leq 2(\|A\| \Upsilon^2 + \mathbb{E}_{s \sim \nu^\pi} \max_{a \in A} (\tilde{Q}_{k+1})^2 + \Upsilon^2)
\]

Thus complete the proof. \( \square \)

We then derive a difference term associated with \( \pi_{k+1} \) and \( \pi_{\theta_k} \), where at the \( k \)-th iteration \( \pi_{k+1} \) is the solution for the following subproblem,

\[
\pi_{k+1} = \arg \max_{\pi} \left( \mathbb{E}_{s \sim \nu_{\pi_k}} [\mathbb{E}_{a \sim \pi} (\tilde{Q}_{\pi, \hat{y}_k}) - \beta \text{KL}(\pi \| \pi_{\theta_k})] \right)
\]

and \( \pi_{\theta_k} \) is the policy parameterized by the two-layered over-parameterized neural network. The following lemma establishes the one-step descent of the KL-divergence in the policy space:

**Lemma 15.** (One-step difference of \( \pi \)) For \( \pi_{k+1} \) and \( \pi_{\theta_k} \), we have

\[
\text{KL}(\pi^* \| \pi_{\theta_k}) - \text{KL}(\pi^* \| \pi_{\theta_{k+1}}) \\
\geq (\mathbb{E}_{a \sim \pi^*} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] - \mathbb{E}_{a \sim \pi_{\theta_k}} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})]) \\
+ \beta^{-1} (\mathbb{E}_{a \sim \pi^*} [Q_{\pi, \hat{y}_k}] - \mathbb{E}_{a \sim \pi_{\theta_k}} [Q_{\pi, \hat{y}_k}]) \\
+ \frac{1}{2} \|\pi_{\theta_{k+1}} - \pi_{\theta_k}\|_2^2 + (\mathbb{E}_{a \sim \pi_{\theta_k}} [\tau_{k+1}^{-1} f_{\theta_{k+1}} - \tau_k^{-1} f_{\theta_k}]) \\
- \mathbb{E}_{a \sim \pi_{\theta_{k+1}}} [\tau_{k+1}^{-1} f_{\theta_{k+1}} - \tau_k^{-1} f_{\theta_k}])
\]

**Proof.** We start from

\[
\text{KL}(\pi^* \| \pi_{\theta_k}) - \text{KL}(\pi^* \| \pi_{\theta_{k+1}}) = \mathbb{E}_{a \sim \pi^*} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] - \mathbb{E}_{a \sim \pi_{\theta_k}} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})]
\]

(By definition, \( \text{KL}(\pi_{\theta_{k+1}} \| \pi_{\theta_k}) = \mathbb{E}_{a \sim \pi_{\theta_k}} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] \))

\[
= (\mathbb{E}_{a \sim \pi^*} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] - \mathbb{E}_{a \sim \pi_{\theta_k}} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})]) + \\
\text{KL}(\pi_{\theta_{k+1}} \| \pi_{\theta_k})
\]

We then add and subtract some terms,

\[
= \mathbb{E}_{a \sim \pi^*} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] - \mathbb{E}_{a \sim \pi_{\theta_k}} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] + \\
(\pi_{\theta_{k+1}} \| \pi_{\theta_k}) + \beta^{-1} (\mathbb{E}_{a \sim \pi^*} [Q_{\pi, \hat{y}_k}] - \mathbb{E}_{a \sim \pi_{\theta_k}} [Q_{\pi, \hat{y}_k}]) \\
- \beta^{-1} (\mathbb{E}_{a \sim \pi^*} [Q_{\pi, \hat{y}_k}] - \mathbb{E}_{a \sim \pi_{\theta_k}} [Q_{\pi, \hat{y}_k}]) + \beta^{-1} (\mathbb{E}_{a \sim \pi^*} [Q_{\pi, \hat{y}_k}] - \mathbb{E}_{a \sim \pi_{\theta_k}} [Q_{\pi, \hat{y}_k}]) + KL
\]

Rearrange the terms and we get,

\[
= (\mathbb{E}_{a \sim \pi^*} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] - \log(\pi_{\theta_k}) - \beta^{-1} \hat{Q}_{\pi, \hat{y}_k}) \\
- \mathbb{E}_{a \sim \pi_{\theta_k}} [\log(\frac{\pi_{\theta_{k+1}}}{\pi_{\theta_k}})] - \log(\pi_{\theta_k}) - \beta^{-1} \hat{Q}_{\pi, \hat{y}_k}) + \beta^{-1} (\mathbb{E}_{a \sim \pi^*} [Q_{\pi, \hat{y}_k}] - \mathbb{E}_{a \sim \pi_{\theta_k}} [Q_{\pi, \hat{y}_k}]) + KL
\]

Recall that \( \pi_{k+1} \propto \exp\left\{ \tau_k^{-1} f_{\theta_k} + \beta^{-1} \frac{\hat{y}}{\tilde{Q}} \right\} \). We define the two normalization factors associated with ideal improved policy \( \pi_{k+1} \) and the current parameterized policy \( \pi_{\theta_k} \) as,

\[
Z_{k+1}(s) := \sum_{a' \in A} \exp\left\{ \tau_k^{-1} f_{\theta_k}(s, a') + \beta^{-1} \frac{\hat{y}}{\tilde{Q}} \right\}
\]

\[
Z_{\theta_k}(s) := \sum_{a' \in A} \exp\left\{ \tau_k^{-1} f_{\theta_k}(s, a') \right\}
\]

We then have,

\[
\pi_{k+1}(a \| s) = \frac{\exp\left\{ \tau_k^{-1} f_{\theta_k}(s, a) + \beta^{-1} \frac{\hat{y}}{\tilde{Q}} \right\}}{Z_{k+1}(s)};
\]

\[
\pi_{\theta_k}(a \| s) = \frac{\exp\left\{ \tau_k^{-1} f_{\theta_k}(s, a) \right\}}{Z_{\theta_k}(s)}
\]

For any \( \pi, \pi' \) and \( k \), we have,

\[
\mathbb{E}_{a \sim \pi} [\log Z_{\theta_{k+1}}] - \mathbb{E}_{a \sim \pi'} [\log Z_{\theta_{k+1}}] = 0
\]

\[
\mathbb{E}_{a \sim \pi} [\log Z_{k+1}] - \mathbb{E}_{a \sim \pi'} [\log Z_{k+1}] = 0
\]

Now we look back at some of the terms on RHS from
For Eq. (C.2), we obtain the first equality by Eq. (C.2). Then, by swapping Eq. (C.2) with Eq. (C.2), we obtain the second equality. We achieve the concluding step with the definition in Eq. (C.2). Following a similar logic, we have,

\[
\begin{align*}
E_{a \sim \pi} \left[ \log (\pi_{\theta_k} + \beta^{-1} Q_{\pi_k, \hat{y}}) \right] \\
= E_{a \sim \pi} \left[ \log (\pi_{\theta_k}) + \beta^{-1} Q_{\pi_k, \hat{y}} - \log Z_{\theta_k} \right] + E_{a \sim \pi} \left[ \log \left( \frac{Z_{\theta_k}}{Z_{\theta_k} + 1} \right) \right] \\
= E_{a \sim \pi} \left[ \log \left( \frac{Z_{\theta_k}}{Z_{\theta_k} + 1} \right) \right] - E_{a \sim \pi} \left[ \log \left( \frac{Z_{\theta_k}}{Z_{\theta_k} + 1} \right) \right] \\
= E_{a \sim \pi} \left[ \log \frac{Z_{\theta_k}}{Z_{\theta_k} + 1} \right]
\end{align*}
\]

Finally, by using the Pinsker’s inequality (Csiszár and Körner, 2011), we have,

\[
\text{KL}(\pi_{\theta_{k+1}} \| \pi_{\theta_k}) \geq 1/2 ||\pi_{\theta_{k+1}} - \pi_{\theta_k}||^2_1
\]

Plugging Eqs. (C.2), (C.2), and (C.2) into Eq. (C.2), we have

\[
\begin{align*}
\text{KL}(\pi^* \| \pi_{\theta_k}) = \text{KL}(\pi^* \| \pi_{\theta_{k+1}}) \\
\geq (E_{a \sim \pi} [\log (\pi_{\theta_{k+1}})] - E_{a \sim \pi} [\log (\pi_{\theta_k})]) \\
= \frac{1}{2} ||\pi_{\theta_k} - \pi_{\theta_{k+1}}||^2_1 + (E_{a \sim \pi} [\log \left( \frac{Z_{\theta_k}}{Z_{\theta_k} + 1} \right)] \\
- E_{a \sim \pi} [\log \left( \frac{Z_{\theta_k}}{Z_{\theta_k} + 1} \right)])
\end{align*}
\]

Rearranging the terms, we obtain Lemma 15.

Lemma 15 serves as an intermediate-term for the major result’s proof. We obtain upper bounds by telescoping this term in Theorem 2. Now we are ready to present the proof for Theorem 2.

Proof. First we take expectation of both sides of Eq. (15) with respect to \( s \sim \nu_{\pi^*} \) from Lemma 15 and insert Eq (13) to obtain,

\[
\begin{align*}
E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_{k+1}}) \right] - E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
\leq \epsilon_k - \beta^{-1} E_{s \sim \nu_{\pi^*}} \left[ ||\pi_{\theta_{k+1}} - \pi_{\theta_k}||^2_1 \right] - E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
= \frac{1}{2} E_{s \sim \nu_{\pi^*}} \left[ ||\pi_{\theta_{k+1}} - \pi_{\theta_k}||^2_1 \right] - E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
\end{align*}
\]

Then, by Lemma 7, we have,

\[
\begin{align*}
\beta^{-1} E_{s \sim \nu_{\pi^*}} \left[ ||\pi_{\theta_{k+1}} - \pi_{\theta_k}||^2_1 \right] - E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
= \beta^{-1} \left( 1 - \gamma \right) \left( J^{\theta_k}_H(\pi^*) - J^{\theta_k}_H(\pi) \right)
\end{align*}
\]

And with Hölder’s inequality, we have,

\[
\begin{align*}
E_{s \sim \nu_{\pi^*}} \left[ ||\pi_{\theta_{k+1}} - \pi_{\theta_k}||^2_1 \right] - E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
\leq \epsilon_k - (1 - \gamma) E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
+ J^{\theta_k}_H(\pi^*) + 1/2 E_{s \sim \nu_{\pi^*}} \left[ ||\pi_{\theta_{k+1}} - \pi_{\theta_k}||^2_1 \right]
\end{align*}
\]

The second inequality holds by using the inequality \( 2pq - q^2 \leq p^2 \), with a minor abuse of notations. Here, \( p := ||\pi_{\theta_{k+1}} - \pi_{\theta_k}||^2_1 \) and \( q := ||\pi_{\theta_k} - \pi_{\theta_{k+1}}||_1 \). Then, by plugging in Lemma 8 and Eq. (14) we end up with,

\[
\begin{align*}
E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_{k+1}}) \right] - E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
\leq \epsilon_k - (1 - \gamma) E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] \\
+ (1 - \gamma) \beta^{-1} \left( \frac{2\epsilon_k M (1 - \gamma) \lambda}{\sqrt{k}} \right) + (\epsilon'_k + \beta^{-2} U)
\end{align*}
\]

Rearrange Eq. (C.2), we have

\[
(1 - \gamma) \beta^{-1} \left( J^{\theta_k}_H(\pi^*) - J^{\theta_k}_H(\pi) \right) \\
\leq E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_k}) \right] - E_{s \sim \nu_{\pi^*}} \left[ \text{KL}(\pi^* \| \pi_{\theta_{k+1}}) \right] \\
+ \left( \frac{2\epsilon_k M (1 - \gamma) \lambda}{\beta \sqrt{k}} \right) + \epsilon_k + \epsilon'_k + \beta^{-2} U
\]
And then telescoping Eq. (C.2) results in,

\[
(1 - \gamma) \sum_{k=1}^{K} \beta^{-1} \min_{k \in [K]} (J^y_\lambda (\pi^*) - J^{\tilde{y}_k}_\lambda (\pi_k)) 
\leq (1 - \gamma) \sum_{k=1}^{K} \beta^{-1} (J^y_\lambda (\pi^*) - J^{\tilde{y}_k}_\lambda (\pi_k)) 
\leq \mathbb{E}_{s \sim \nu_s} [\text{KL}(\pi^* || \pi_0)] - \mathbb{E}_{s \sim \nu_s} [\text{KL}(\pi^* || \pi_K)] 
+ \lambda r_{\text{max}} (1 - \gamma)^2 \sum_{k=1}^{K} \beta^{-1} \left( \frac{2c_3}{\sqrt{K}} \right) + U \sum_{k=1}^{K} \beta_k^{-2} 
+ \sum_{k=1}^{K} (\epsilon_k + \epsilon'_k)
\]

We complete the final step in Eq. (C.2) by plugging in Lemma 8 and Eq. (13). Per the observation we make in the proof of Theorem 1,

1. \( \mathbb{E}_{s \sim \nu_s} [\text{KL}(\pi^* || \pi_0)] \leq \log \mathcal{A} \) due to the uniform initialization of policy.
2. \( \text{KL}(\pi^* || \pi_K) \) is a non-negative term.

We now have,

\[
\min_{k \in [K]} J^y_\lambda (\pi^*) - J^{\tilde{y}_k}_\lambda (\pi_k) 
\leq \log |\mathcal{A}| + UK \beta^{-2} \sum_{k=1}^{K} (\epsilon_k + \epsilon'_k) 
+ \lambda r_{\text{max}} (1 - \gamma)^2 \left( \frac{2c_3}{\sqrt{K}} \right) 
\]

Replacing \( \beta \) with \( \beta_0 \sqrt{K} \) finishes the proof.

D. Acknowledgement

The time we released the manuscript on arxiv (01/24/2022), we realized (through personal communications) that a paper of possible similar research topic titled “Finite Sample Analysis of Mean-Volatility Actor-Critic for Risk-Averse Reinforcement Learning” is accepted by AISTATS’2022 a few days ago, and has not been released in any form online yet (arxiv, author’s personal webpage, etc.). It should be noted that this work is in parallel with the aforementioned paper.