Tilted Quantile Gradient Updates for Quantile-Constrained Reinforcement Learning

Chenglin Li[1], **Guangchun Ruan**[2], **Hua Geng**[1]*

[1]Department of Automation, Tsinghua University, Beijing, China
[2]Laboratory for Information & Decision Systems, MIT, Boston, USA
licl23@mails.tsinghua.edu.cn, gruan@mit.edu, genghua@tsinghua.edu.cn

Abstract

Safe reinforcement learning (RL) is a popular and versatile paradigm to learn reward-maximizing policies with safety guarantees. Previous works tend to express the safety constraints in an expectation form due to the ease of implementation, but this turns out to be ineffective in maintaining safety constraints with high probability. To this end, we move to the quantile-constrained RL that enables a higher level of safety without any expectation-form approximations. We directly estimate the quantile gradients through sampling and provide the theoretical proofs of convergence. Then a tilted update strategy for quantile gradients is implemented to compensate the asymmetric distributional density, with a direct benefit of return performance. Experiments demonstrate that the proposed model fully meets safety requirements (quantile constraints) while outperforming the state-of-the-art benchmarks with higher return.

Introduction

There has been rising attention on Safe reinforcement learning (RL) which seeks to develop a policy that maximizes the expected cumulative return while meeting all the safety constraints. A common option is the Constrained Markov Decision Process (CMDP) framework, in which the safety constraints are established to bound the expected cumulative cost $C = \sum_t \gamma^t c$ by:

$$\mathbb{E}[C] \leq d \tag{1}$$

Most previous works follow this framework to construct constraints, because this form is consistent with the expected cumulative return (objective function) such that the gradient calculation would become the same for both the safety constraints and the objective function. As for the solution, the Lagrange approach is common (Achiam et al. 2017; Liang, Que, and Modiano 2018; Tessler, Mankowitz, and Mannor 2018; Liu, Ding, and Liu 2020), but there are other options such as projection-based methods (Yang et al. 2020), shielding methods (Alshiekh et al. 2018; Carr et al. 2023), and barrier methods (Marvi and Kiumarsi 2021; Cheng et al. 2019).

Unfortunately, the above expectation setup does not apply for many real-world safety-critical applications. Bounding expectations may still induce constraint violation in extreme cases, and the potential risk cannot be strictly limited. A better solution is to apply probability-related constraints, e.g. 95%-quantile, to impose a more accurate and robust requirement for safety. In this case, the probability $\Pr[C \leq d]$ can be more informative than the safety expectation $\mathbb{E}[C] \leq d$.

A probabilistic constraint or a chance constraint is typically expressed as follows (Chow et al. 2018; Chen, Subramanian, and Paternain 2024):

$$\Pr[C \leq d] \geq 1 - \varepsilon \tag{2}$$

Eqn. (2) strictly limits the probability of constraint violation under a given level ε, but this constraint is computationally intractable and suffers from low sample efficiency and lack of distribution priors. A direct transformation of (2) needs to apply the quantile, or value-at-risk (VaR) metric. Mathematically, Eqn. (2) is equivalent to the following quantile constraint:

$$q_{1-\varepsilon} := \inf\{x | \Pr[C \leq x] \geq 1 - \varepsilon\} \leq d \tag{3}$$

An optimization model with quantile constraints integrated is still computationally intensive for training. In the literature, quantile optimization has been widely studied. Estimating the gradient of quantile is a challenge, and most of the existing solutions such as the perturbation analysis (Jiang and Fu 2015), the likelihood ratio method (Glynn et al. 2021), and the kernel density estimation (Hong and Liu 2009) are heavily relied on the analytical model formulation. Also, these models are focused on unconstrained optimization problems with quantile-based objectives, but the focus of this paper is on the quantile constrains for Safe RL.

Quantile-constrained RL was first studied in (Jung et al. 2022), and the idea was to supplement the expected cumulative cost $\mathbb{E}[C]$ with an additive term to approximate the quantile $q_{1-\varepsilon}$. Within this setting, the quantile constraint could be converted into an expectation-type constraint at last, aligning with the CMDP framework. Note that this work required the cumulative cost distribution, and its empirical performances might be over-conservative (with relatively low return) because of the biased approximation of quantiles.

Other related works have also been conducted to bound the probability of constraint violation. Yang et al. (2023) applied Conditional Value-at-Risk (CVaR) as an approximation of the quantile. Chow et al. (2018) proposed

*Corresponding author.

a trajectory-based method with chance constraints to bound the probability of constraint violation. However, this trajectory-based approach updates the policy only once based on a batch of trajectories, resulting in low sample efficiency, which is not suitable for practical application. Some model-based methods have also been proposed to guarantee the safety probability (Peng et al. 2022; Pfrommer et al. 2022), but these method requires prior knowledge of the environment, which is not practical in many real-world scenarios.

In this paper, we establish a novel quantile-constrained RL model, namely Tilted Quantile Policy Optimization (TQPO) model, where the safety constraints are expressed in a quantile form as Eqn. (3). For the estimation of quantile gradients, we get rid of any expectation-form approximations and directly estimate the quantile gradient through a sampling technique. To avoid over-conservatism of the policy and gain higher return, a tilted quantile gradient update is designed to compensate the asymmetric distributional density of quantiles.

The rest of this paper is structured as follows. We first introduce the background in the Preliminaries section. Then we present the methodology of the TQPO model, followed by a convergence analysis. In the Experiments section, simulation results demonstrate that the proposed model fully guarantee safety while outperforming the state-of-the-art benchmarks with higher returns.

Preliminaries

Constrained Markov Decision Process

A CMDP framework (Altman 2021) is characterized by a tuple $\langle S, A, P, r, c, d, \gamma \rangle$, where S denotes the state space, A denotes the action space, $P(\cdot|s,a)$ is the state transition probability function, $r(s,a)$ is the reward function, $c(s,a)$ is the cost function, d is a given threshold, and $\gamma \in (0,1)$ is the discount factor.

The agent interacts with the environment at each time step t by observing the current state $s_t \in S$ and selecting an action $a_t \in A$, then receives a reward $r(s_t, a_t)$ as well as a cost $c(s_t, a_t)$. This agent focuses on learning a policy $\pi_\theta(\cdot|s)$ parameterized by θ. The next states are generated by the state transition probability function $P(\cdot|s,a)$ and the policy $\pi_\theta(\cdot|s)$. Given an initial state s_0, the cumulative return is defined as $R(s_0, \pi_\theta) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$, and the cumulative cost is defined as $C(s_0, \pi_\theta) = \sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)$.

Based on the previous definitions, CMDP framework is established to find a policy π_θ that maximizes the expected return while satisfying the expected cost under a given threshold d:

$$\begin{aligned} &\max_\theta V(s, \pi_\theta) := \mathbb{E}_{\pi_\theta}[R(s, \pi_\theta)] = \mathbb{E}_{\pi_\theta}[\sum_t \gamma^t r(s_t, a_t)] \\ &\text{s.t. } \mathbb{E}_{\pi_\theta}[C(s, \pi_\theta)] = \mathbb{E}_{\pi_\theta}[\sum_t \gamma^t c(s_t, a_t)] \le d \end{aligned} \tag{4}$$

where $V(s, \pi_\theta)$ denotes the state value function. The expectation term of constraints matches the same term of the objective, therefore the gradient of the constraint can be calculated in a similar way, resulting in easy implementation by the Lagrangian method. However, the expectation-based formulation of the constraints cannot strictly limit the probability of constraint violation, which is not suitable for safety-critical applications.

Quantile-constrained RL

For a sequence of samples $\{s_0, s_1, \ldots, s_N\}$, the cumulative cost $C(s_i, \pi_\theta)$ of each state s_i follows the empirical distribution $F(\cdot; \pi_\theta)$. Given a probability level $\varepsilon \in (0,1)$, similar to Eqn. (3) with the cumulative cost $C(s, \pi_\theta)$ following the distribution $F(\cdot; \pi_\theta)$, the $1-\varepsilon$ quantile of $C(s, \pi_\theta)$ can be rewritten as:

$$q_{1-\varepsilon}(\pi_\theta) := \inf\{q | \Pr(C(s, \pi_\theta) \le q) = F(q; \pi_\theta) \ge 1-\varepsilon\} \tag{5}$$

When the quantile of the distribution of C satisfies $q_{1-\varepsilon}(\pi_\theta) \le d$, the probability of constraint violation is under ε. Therefore, the quantile-constrained RL problem can be formulated as follows:

$$\begin{aligned} &\max_\theta V(s, \pi_\theta) = \mathbb{E}_{\pi_\theta}[R(s, \pi_\theta)] = \mathbb{E}_{\pi_\theta}[\sum_t \gamma^t r(s_t, a_t)] \\ &\text{s.t. } q_{1-\varepsilon}(\pi_\theta) \le d \end{aligned} \tag{6}$$

Since the quantile $q_{1-\varepsilon}(\pi_\theta)$ is not of an expectation form, the conventional expectation Bellman equation cannot solve the gradient calculation of the quantile constraint. How to estimate quantile gradients become a challenge in this problem setting.

Methodology

In this section, we use a sample-based approach to estimate the gradient of the quantile constraint, and then construct a tilted quantile gradient update to accelerate the training process. Finally, we proposed a quantile-constrained RL algorithm based on the tilted quantile gradient update.

Estimating Quantile Gradients Through Sampling

Consider a quantile constraint as Eqn. (5). When $F(\cdot; \pi_\theta)$ is continuous and differentiable (a minor assumption), the quantile $q_{1-\varepsilon}(\pi_\theta)$ is mathematically the inverse of $F(\cdot; \pi_\theta)$. According to the inverse function theorem, the gradient of this quantile constraint can be calculated as follows:

$$\nabla_\theta q_{1-\varepsilon}(\pi_\theta) = \nabla_\theta F^{-1}(1-\varepsilon; \pi_\theta) = -\frac{\nabla_\theta F(q; \pi_\theta)}{f(q; \pi_\theta)}\Big|_{q=q_{1-\varepsilon}(\pi_\theta)} \tag{7}$$

where $f(\cdot; \pi_\theta)$ is the probability density function (PDF) of the cumulative cost C. Eqn. (7) implies that the quantile gradient can be estimated by figuring out the above numerator and denominator.

The numerator $\nabla_\theta F(q; \pi_\theta)$ can be estimated by the likelihood ratio method in policy gradient algorithms. Given a batch of samples $\{s_0, s_1, \ldots s_N\}$, the gradient of the CDF

can be estimated as follows:

$$\begin{aligned}\nabla_\theta F(q;\pi_\theta) =& \nabla_\theta \mathbb{E}[I(C(s,\pi_\theta)\geq q)] = -\nabla_\theta \mathbb{E}[I(C(s,\pi_\theta)\leq q)] \\ =& -\mathbb{E}[I(C(s,\pi_\theta)\leq q)\sum_{t=i}^{N-1}\nabla_\theta \log \pi_\theta(a_t|s_t)] \\ \approx& -\frac{1}{N}\sum_{i=1}^{N} I(C(s_i,\pi_\theta)\leq q)\sum_{t=i}^{N-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\end{aligned} \tag{8}$$

where $q = q_{1-\varepsilon}(\pi_\theta)$ denotes the $1-\varepsilon$ quantile, which is unknown in practice. We adopt a iterative method to estimate the quantile $q_{1-\varepsilon}(\pi_\theta)$ with q_k from every batch of samples as follows:

$$q_{k+1} = q_k + \alpha(\hat{q}_{1-\varepsilon} - q_k) \tag{9}$$

where $\hat{q}_{1-\varepsilon}$ is the empirical quantile of the cumulative cost C from the batch of samples, and $\alpha \in (0,1)$ is the update rate for smoothness.

The denominator $f(q;\pi_\theta)$ in Eqn. (7) is the PDF of the cumulative cost C, which is difficult to estimate without the priori of the environment (Jiang, Peng, and Hu 2022). Since PDF is always positive, the gradient of quantile in Eqn. (7) is in the same direction as $-\nabla_\theta F(q;\pi_\theta)|_{q=q_{1-\varepsilon}(\pi_\theta)}$, which can be adopted as an approximation of the gradient of the quantile constraint.

Collectively, the quantile gradient $\nabla_\theta q_{1-\varepsilon}(\pi_\theta)$ can be estimated by applying Eqn. (7)–(9). Therefore, policy gradient method is applicable solve the quantile-constrained RL problem in Eqn. (6). Applying the Lagrangian method, the dual objective of the quantile-constrained RL problem is defined as follows:

$$\min_{\lambda\geq 0}\max_{\theta} \mathcal{L}(\theta,\lambda,q) = V(s,\pi_\theta) - \lambda(q_{1-\varepsilon}(\pi_\theta) - d) \tag{10}$$

Given a batch of samples $\{s_0, s_1, \ldots, s_N\}$, the gradient of the dual objective in Eqn. (10) w.r.t. policy parameter θ can be calculated as follows. This gradient can then be used to update the policy parameter θ:

$$\begin{aligned}&\nabla_\theta \mathcal{L}(\theta,\lambda,q) = \nabla_\theta V(s,\pi_\theta) - \lambda\nabla_\theta q_{1-\varepsilon}(\pi_\theta) \\ \approx& \frac{1}{N}\sum_{i=1}^{N}\Big[V(s_i,\pi_\theta) - \lambda I(C(s_i,\pi_\theta)\leq q_k)\Big]\sum_{t=i}^{N-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\end{aligned} \tag{11}$$

For the Lagrangian multiplier λ, it can be updated by the gradient of the dual objective w.r.t. λ as follows:

$$\nabla_\lambda \mathcal{L}(\theta,\lambda,q) = -(q_{1-\varepsilon}(\pi_\theta) - d) \tag{12}$$

Then the Lagrangian multiplier λ can be updated as follows:

$$\lambda \leftarrow \max\{\lambda + \eta(q_{1-\varepsilon}(\pi_\theta) - d), 0\} \tag{13}$$

where η is the update rate of λ.

Tilted Quantile Gradient Update

However, a direct use of Eqn. (13) with a fixed η to update λ can be inefficient due to the overshooting of λ at the early stage of training. The reason is that the quantile $q_{1-\varepsilon}(\pi_\theta)$ may have an asymmetric distributional density around the threshold d. We illustrate this issue as follows.

Assuming a given violation probability level ε, at the early stage of training, the initial policy may behave unsafe with violation probability much larger than ε, as well as $q_{1-\varepsilon}(\pi_\theta)$ several times larger than the threshold d, resulting in a large increase of λ. This large increase makes λ overshoot to a large value at the early training, which may result in over-conservatism of the policy (Peng et al. 2022).

Later when the policy satisfies the constraint with $0 \leq q_{1-\varepsilon}(\pi_\theta) \leq d$, λ starts to decrease with a slower rate. Even for an absolute safe policy with $q_{1-\varepsilon}(\pi_\theta) = 0$ where λ decreases fastest, the decrease of λ is still slow as $\Delta\lambda = \eta(q_{1-\varepsilon}(\pi_\theta) - d) = -\eta d$, rather than the rapid increase earlier with $q_{1-\varepsilon}(\pi_\theta)$ several times larger than d. The slow decrease of λ from a large value may result in a slow recovery of the policy from over-conservatism. In general, this asymmetric distributional density of the quantile indicates that $q_{1-\varepsilon}(\pi_\theta) - d$ is relatively large in the early stage of training, but not small enough at the later stage. This issue makes λ overshoot rapidly at the early unsafe training, while decrease slowly at the later over-conservatism stage, which may eventually result in slow convergence of the algorithm.

To address this issue, we propose a tilted quantile gradient update to compensate the asymmetric distributional density of $q_{1-\varepsilon}(\pi_\theta)$. Since we expect the distribution of $q_{1-\varepsilon}(\pi_\theta)$ to be symmetric around d, a tilted factor is designed to compensate the asymmetric distribution. Similar to the pinball loss in quantile regression (Steinwart and Christmann 2011), we revise the update rate η in Eqn. (13) with a tilted term defined as follows:

$$\eta = \begin{cases} \eta_+ = \frac{F_q(d)+\delta}{1+\delta}, & \text{if } q_{1-\varepsilon}(\pi_\theta) \geq d \\ \eta_- = \frac{1-F_q(d)+\delta}{1+\delta}, & \text{if } q_{1-\varepsilon}(\pi_\theta) < d \end{cases} \tag{14}$$

where $F_q(d)$ denotes the CDF of the distribution of quantile $q_{1-\varepsilon}(\pi_\theta)$ at d, which is estimated by sampling per epoch. η_+ and η_- are the update rates for the positive and negative tilted terms respectively, $\delta \in (0,1)$ a small smoothing factor.

The tilted term in Eqn. (14) is utilized to update the Lagrangian multiplier λ in Eqn. (13). For example, in the early stage of training, the policy is always unsafe with $q_{1-\varepsilon}(\pi_\theta) \geq d$, resulting in a small positive update rate $\eta = \eta_+ \approx \delta$. With the increase of λ, the policy gradually satisfies the constraint with $q_{1-\varepsilon}(\pi_\theta) \leq d$, then λ switches to decrease with a negative update rate $\eta = \eta_- \approx 1-\delta$. Assuming $\delta = 0.1$, the decrease update rate $\eta_- \approx 0.9$ will be about 9 times larger than the increase rate $\eta_+ \approx 0.1$. Therefore, the decrease of λ from a large value can be accelerated, with the tilted term compensating the asymmetric distributional density of $q_{1-\varepsilon}(\pi_\theta)$, which facilitates the recovery of the policy from over-conservatism.

The tilted term in Eqn. (14) is performed each epoch to update λ adaptively, boosts the decrease of λ from overshoot and tunes it to a more appropriate value range eventually, which can prevent the policy from over-conservatism and facilitate it to achieve higher return.

Tilted Quantile Policy Optimization

Based on the quantile gradient estimation with sampling and the tilted quantile gradient update, we propose an algorithm named Tilted Quantile Policy Optimization (TQPO) to solve the quantile-constrained RL problem in Eqn. (6). The algorithm is based on the classic RL algorithm Proximal Policy

Optimization (PPO) (Schulman et al. 2017) with the quantile constraint. We use a policy network parameterized by θ to represent the policy $\pi_\theta(\cdot|s)$, and a value network parameterized by ϕ to obtain the estimated value function $V_\phi(s)$.

The loss function of θ is defined as follows to train the policy network:

$$L_\theta = -\mathbb{E}_\pi\Big[\min\Big(\frac{\pi(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}A, \text{clip}(\frac{\pi(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}, 1-r_{clip}, 1+r_{clip})A\Big)\Big] \tag{15}$$

where

$$A = r(s_i,a_i) + \gamma V_\phi(s_{i+1},\pi_\theta) - V_\phi(s_i,\pi_\theta) - \lambda I(C(s_i,\pi_\theta) \le q_k) \tag{16}$$

Eqn. (16) is the reward advantage function in PPO style, with the addition of the quantile constraint gradient. Additionally, we adopt the importance sampling technique and clipped surrogate objective in PPO, shown in Eqn. (15), to stabilize the training and improve sample efficiency.

Overall, the training process of TQPO iterates as follows:

- Generate a batch of samples $\{s_0, s_1, \ldots, s_N\} \sim \pi_\theta$
- Update the value network parameter ϕ
- Update three main parameters q, θ and λ as follows:

$$q_{k+1} = q_k + \alpha_k(\hat{q}_{1-\varepsilon} - q_k) \tag{17a}$$
$$\theta_{k+1} = \theta_k + \beta_k \nabla_\theta L_\theta \tag{17b}$$
$$\lambda_{k+1} = \lambda_k + \eta_k(q_k - d) \tag{17c}$$

where α_k, β_k, and η_k are the update rates of the three parameters respectively. Implementation details of TQPO can be found in Appendix B.

Convergence Analysis

In this section, we provide the theoretical proofs of the convergence of the TQPO algorithm. First, let's reconsider the update of the three main parameters in the TQPO algorithm, i.e., the estimated quantile q, the policy parameter θ and the Lagrange multiplier λ in Eqn. (17). For the convenience of theoretical analysis, We modify Eqn. (17b) by replacing the implemented loss L_θ to the Lagrange objective function $\mathcal{L}(\theta,\lambda,q)$ in Eqn. (10). In order to prove the convergence, we first adopt the following assumptions:

Assumption 1. *For any probability level $\varepsilon \in (0,1)$, the objective function $\mathcal{L}(\theta,\lambda,q)$ is continuous and differentiable with respect to θ.*

Assumption 2. *$\nabla_\theta\mathcal{L}(\theta,\lambda,q)$ is Lipschitz continuous w.r.t. θ, λ and q, i.e., $\forall(\theta_1,\lambda_2,q_2),(\theta_2,\lambda_2,q_2) \in \Theta\times\mathbb{R}_+\times\mathbb{R}$, there exists a constant κ such that $\|\nabla_\theta\mathcal{L}(\theta_1,\lambda_1,q_1) - \nabla_\theta\mathcal{L}(\theta_2,\lambda_2,q_2)\| \le \kappa\|(\theta_1,\lambda_1,q_1) - (\theta_2,\lambda_2,q_2)\|$, where Θ is the parameter space of the policy network.*

Assumption 3. *The update rates α_k, β_k, and η_k are all positive, nonsummable, and square summable. In specific, this indicates that $\alpha_k > 0, \sum_{k=0}^\infty \alpha_k = \infty, \sum_{k=0}^\infty \alpha_k^2 < \infty$. Moreover, the update rates satisfy: $\eta_k = o(\beta_k), \beta_k = o(\alpha_k)$.*

Assumption 1 is a common condition in continuous optimization, which ensures the continuity of the objective function w.r.t the policy parameter θ. Assumption 2 and 3 are standard conditions for stochastic approximation analysis (Borkar 2008; Gattami, Bai, and Aggarwal 2021).

Assumption 3 indicates the update timescales of the quantile q, the policy parameter θ, and the Lagrange multiplier λ, respectively, where q is updated fastest, followed by θ, and λ is the slowest. The three parameters affect each other in the updating but with different timescales. Therefore, we can utilize the timescale separation to conduct the proof by two steps, first proving the convergence of (θ,q), and then proving the convergence of (θ,q,λ).

Convergence of (θ, q)

First, we consider the convergence of the quantile q and the policy parameter θ in the TQPO algorithm. Since λ updates slower than q and θ, we can regard λ as an arbitrary constant in the timescale of q and θ.

Considering the updates of q and θ in Eqn. (17a) and Eqn. (17b), the two recursions are expected to track two coupled odinary differential equations (ODEs) with respect to q and θ:

$$\dot{q}(t) = g_1(\theta,q) := \hat{q}_{1-\varepsilon}(\theta) - q$$
$$\dot{\theta}(t) = g_2(\theta,q) := \nabla_\theta\mathcal{L}(\theta,q) \tag{18}$$

Since the update rate of θ is slower than q, we can regard θ as a constant when updating q.

Lemma 1. *For any $\overline{\theta} \in \Theta$, the ODE $\dot{q}(t) = g_1(\overline{\theta},q)$ has the unique global asymptotically stable equilibrium $q_{\overline{\theta}}$.*

With the support of Lemma 1, $\{q_k\}$ converges to the equilibrium of the ODE $\dot{q}(t) = g_1(\theta,q)$ for any $\theta \in \Theta$. Then we focus on the convergence of θ. The gradient update of θ in Eqn. (17b) can be considered as tracking the right-hand side of the ODE in Eqn. (18). Therefore, we adopt the following theorem to prove $\{\theta_k\}$ converge to the unique global asymptotically stable equilibrium.

Theorem 1. *For the two coupled iterations:(Borkar 1997)*

$$q_{k+1} = q_k + \alpha_k(g_1(\theta_k,q_k) + m_k)$$
$$\theta_{k+1} = \theta + \beta_k(g_2(\theta_k,q_k) + n_k) \tag{19}$$

for $k \ge 0$, where,

- *(i): $g_1(\theta,q)$ and $g_2(\theta,q)$ are Lipschitz continuous*
- *(ii): α_k and β_k satisfy Assumption 3*
- *(iii): m_k and n_k are noise sequences and satisfy $\sum_{k=0}^\infty \alpha_k m_k,\ \sum_{k=0}^\infty \beta_k n_k < \infty$*

If $\forall\overline{\theta} \in \Theta$, ODE $\dot{q}(t) = g_1(\overline{\theta},q)$ has a unique global asymptotically stable equilibrium point $q_{\overline{\theta}}$, the iterations 19 converge to the unique global asymptotically stable equilibrium of the ODE $\dot{\theta}(t) = g_2(\theta,q)$ a.s. on the set $\sup_k|q_k| < \infty$.

Theorem 1 requires $\{q_k\}$ and the log gradient of π_θ to be bounded, which can be guaranteed by the following lemma and assumption respectively.

Lemma 2. *If Assumptions 1, 3 hold, the sequence $\{q_k\}$ satisfies $\sup_k|q_k| < \infty$.*

Assumption 4. *The log gradient of the policy network $\nabla_\theta \log \pi(a|s,\theta)$ is bounded on the state space S w.r.t. the policy parameter $\theta \in \Theta$, i.e., $\sup_{s\in S} \|\nabla_\theta \log \pi(a|s,\theta)\| < \infty$.*

With the support of Lemma 2, Theorem 1 indicates the sequence $\{\theta_k\}$ converges to the unique global asymptotically stable equilibrium of the ODE $\dot{\theta}(t) = g_2(\theta, q)$. So far, we have proved (θ, q) converge to their unique global asymptotically stable equilibriums. We then prove this converged θ is the optimal policy parameter with following lemma:

Lemma 3. *If $\mathcal{L}(\theta, q)$ is strictly concave on Θ, the ODE $\dot{\theta}(t) = g_2(\theta, q)$ has a unique global asymptotically stable equilibrium point $\theta^* = arg\max_\theta \mathcal{L}(\theta, q)$.*

Lemma 3 indicate that the optimal policy parameter θ^* is the unique global asymptotically stable equilibrium point of the ODE $\dot{\theta}(t) = g_2(\theta, q)$.

Above all, we first utilize Lemma 1 and Theorem 1 to prove the convergence of (θ, q) to the unique global asymptotically stable equilibriums of ODEs(18). Then we use Lemma 3 to prove the optimality of the converged θ. Therefore, (θ, q) in TQPO algorithm converges to the optimal (θ^*, q^*) for a fixed λ. Next we provide the convergence analysis of the Lagrange multiplier λ.

Convergence of λ

Numerous works have proved the convergence of the Lagrange multiplier in two timescales constrained MDPs, where (θ, λ) converges to the optimal solution under certain conditions (Paternain et al. 2019; Gattami, Bai, and Aggarwal 2021). As mentioned before, the update rate of λ is slower than both q and θ, it is reasonable to merge the two faster parameters θ and q into a new parameter $\theta' = (\theta, q)$. In the update process of λ, we can regard θ' converged to $\theta'^* = (\theta^*(\lambda), q^*(\lambda))$. Therefore, the three timescales update of (q, θ, λ) can be considered as a two timescales update of (θ', λ), and the standard analysis for constrained MDPs can be applied to the TQPO algorithm.

Theorem 2. *Under Assumptions 1, 2, 3, 4, if Slater's condition holds, the iterates $(\theta'_k, \lambda_k) = (q_k, \theta_k, \lambda_k)$ converge to the optimal solution a.s. (Borkar 2008)*

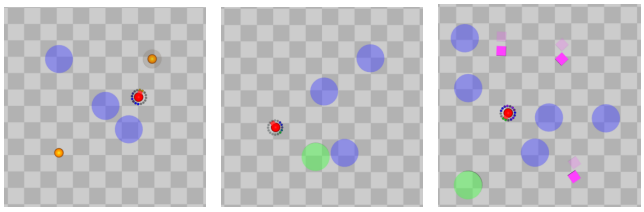
Theorem 2 is a standard analysis for the convergence of the dual problem in constrained MDPs in the RL literature. By Theorem 2, we can conclude that the TQPO algorithm converges to the optimal solution of the quantile-constrained RL problem. Detailed proofs for the lemmas and theorems can be found in the Appendix A.

Experiments

Simulation Setup

We evaluate the proposed TQPO on three classic safe RL tasks: SimpleEnv, DynamicEnv and GremlinEnv from Mujoco and Safety Gym (Todorov, Erez, and Tassa 2012; Ray, Achiam, and Amodei 2019).

In these tasks, a robot (red) is required to reach a goal while avoiding collisions with obstacles. The complexity of the three tasks gradually increases due to the addition of randomness. In SimpleEnv (Fig. 1a), the obstacles include fixed hazards (blue) and none-goal buttons (orange). When the robot reach the goal (orange covered by grey shadow), the environment swaps the goal and the none-goal button, therefore the new goal is generated deterministically. In DynamicEnv (Fig. 1b), when the robot reaches the goal (green), a new goal is generated randomly. In GremlinEnv (Fig. 1c), the obstacles include moving gremlins (pink) and the goal is generated randomly.



(a) SimpleEnv (b) DynamicEnv (c) GremlinEnv

Figure 1: Safety Gym simulation environments

The reward is defined as the distance reduction from the robot to the goal over two time steps. When the robot reaches the goal, the environment provide an additional reward $+1$ and moves the goal to a fixed (Fig. 1a) or random position (Fig. 1b, 1c). When the robot collides with other objects, it receives a cost $+1$ at this time step, otherwise the cost is 0. This reward and cost shaping encourages the robot to reach the goal as many times as possible while avoiding collisions in an episode of 1000 steps.

Baseline Methods and Evaluation

We compare the proposed TQPO with the state-of-the-art quantile-constrained RL algorithm QCPO in (Jung et al. 2022) and a classic expectation-constrained RL method PPO-Lag in (Stooke and Abbeel 2019). Four metrics are used to evaluate the performance of the algorithms: Average episode return, safety probability, average episode cost and $1-\varepsilon$ quantile of cost. The cost threshold is set to $d = 15$. Since many safety-critical applications require a high safety probability above 90%, $1-\varepsilon = 90\%, 95\%$ are used in the experiments. All the experiments are conducted with five random seeds, with the solid line representing the mean and shaded area indicating the standard deviation. Implementation details can be found in Appendix B[1].

Results

First, we prove that compared to the quantile constraint, **the expectation constraint is not suitable for safety-critical scenarios**. Fig. 2 shows the average cost (Row 1) and $1-\varepsilon$ quantile of cost (Row 2). From Row 1, we can observe that PPO-Lag (green) satisfies its expectation constraint $\mathbb{E}[C] \leq d$, with the average cost around the threshold (black line). However, as shown in Row 2, the 90% cost quantile of PPO-Lag significantly exceeds the threshold. In contrast, both QCPO (blue and purple) and TQPO (red and orange) satisfy their quantile constraints $q_{1-\varepsilon}(\pi_\theta) \leq d$, with their cost

[1]Code is available at https://github.com/CharlieLeeeee/TQPO

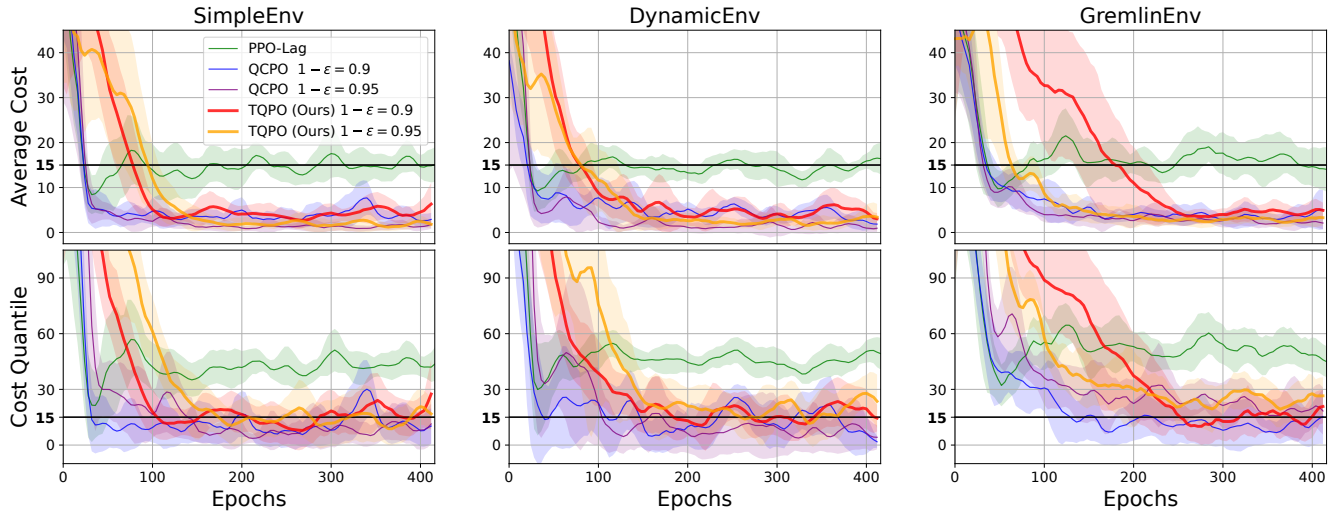


Figure 2: Average Cost (Row 1) and Cost Quantile (Row 2) of three algorithms on SimpleEnv (Column 1), DynamicEnv (Column 2) and GremlinEnv (Column 3). The cost quantile of PPO-Lag is calculated with $1 - \varepsilon = 90\%$

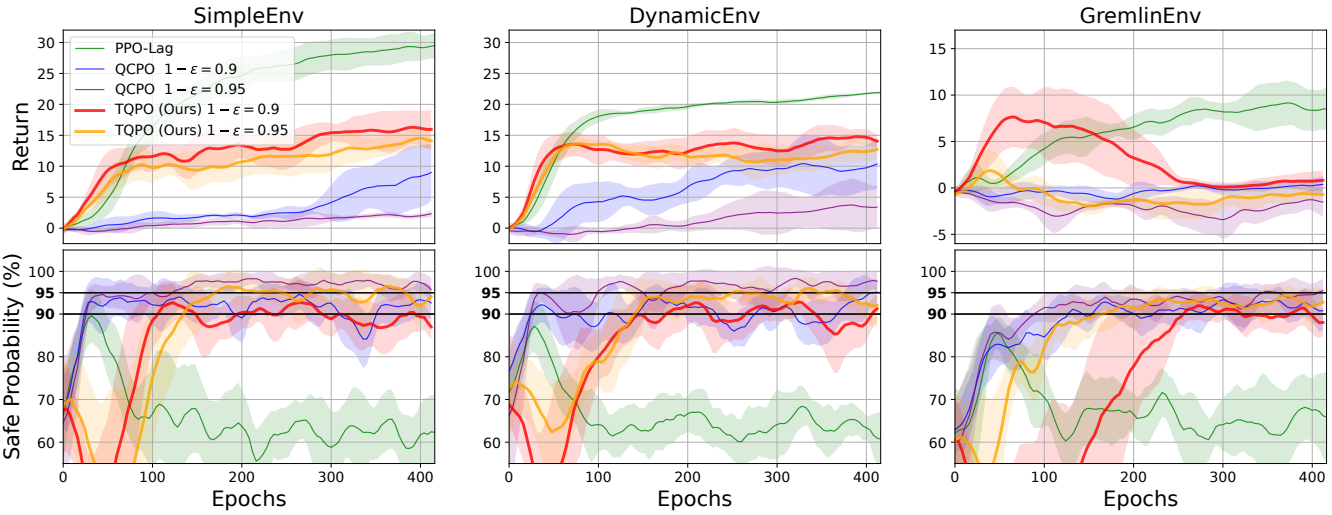


Figure 3: Return (Row 1) and Safety Probability (Row 2) of three algorithms on SimpleEnv (Column 1), DynamicEnv (Column 2) and GremlinEnv (Column 3).

quantiles around the threshold (black line) in Row 2, and their average cost is below that of PPO-Lag in Row 1. This difference in cost between the expectation-based method and the quantile-constrained methods results in the difference in safety probability. Fig. 3 Row 2 demonstrates that the safety probability of PPO-Lag is below 70%, which is significantly lower than that of QCPO and TQPO. As discussed in the introduction, the expectation constraint fails to bound the safety probability, resulting in the low safety probability of PPO-Lag. In comparison, QCPO and TQPO achieve higher safety probabilities closer to the given level (black line) in Fig. 3 Row 2. Therefore, in safety-critical scenarios with high safety probability requirements, the quantile constraint can achieve better safety performance and is more suitable than the expectation-based constraint.

Next, we compare the two quantile-constrained methods. For a more intuitive comparison, we evaluate the average performance of trained QCPO and TQPO algorithms, as shown in Table 1. Higher return and closer safety probability to the given level are preferred, as highlighted by the bolded values.

First consider **the safety probability of QCPO and TQPO**. Fig. 3 Row 2 shows that both QCPO (blue for $1 - \varepsilon = 90\%$, purple for $1 - \varepsilon = 95\%$) and TQPO (red for $1 - \varepsilon = 90\%$, orange for $1 - \varepsilon = 95\%$) achieve safety probability close to the given level $1 - \varepsilon$ (black line). However, the curves of QCPO are more likely to be above $1 - \varepsilon$ rather than around the given probability level like TQPO. Table 1 (Pr Columns) also demonstrates that the safety probability of TQPO is closer to the given level. As mentioned before, QCPO assumes the cumulative cost follows a certain distribution and uses an additive form on $\mathbb{E}[C]$ to ap-

Tasks	Metrics	QCPO	TQPO
SimpleEnv 90%	R	8.6±0.4	**16.1±0.2**
	Pr	92%±1%	**89%±1%**
	T(h)	3.0±0.1	**2.7±0.1**
SimpleEnv 95%	R	2.1±0.2	**14.2±0.3**
	Pr	97%±1%	**95%±1%**
	T(h)	3.2±0.2	**2.6±0.1**
DynamicEnv 90%	R	10.0±0.3	**14.4±0.4**
	Pr	91%±1%	**90%±1%**
	T(h)	2.6±0.2	2.6±0.1
DynamicEnv 95%	R	3.5±0.1	**12.5±0.2**
	Pr	97%±0%	**94%±1%**
	T(h)	**2.7±0.1**	3.0±0.2
GremlinEnv 90%	R	0.33±0.1	**0.77±0.1**
	Pr	92%±1%	**90%±1%**
	T(h)	4.4±0.3	**3.4±0.1**
GremlinEnv 95%	R	-1.43±0.2	**-0.66±0.1**
	Pr	**94%±1%**	93%±1%
	T(h)	4.0±0.2	**3.6±0.2**

Table 1: Empirical results of QCPO and TQPO on three tasks with safety probability $1-\varepsilon = 90\%, 95\%$. R: Average episode return, Pr: Safety probability, T: Training time.

proximate the quantile, which may lead to biased quantile estimation, resulting in higher safety probability. In contrast, TQPO avoids any distribution assumption and expectation-form approximation, directly estimates the quantile through a sampling technique. Results in Fig. 3 Row 2 and Table 1 demonstrate the safety probability of TQPO is closer to the given level, indicating the accuracy of our quantile gradient estimation method.

Next we compare **the return of QCPO and TQPO**. Fig. 3 Row 1 shows that TQPO achieves higher return than QCPO in all tasks, with the return curves of TQPO above those of QCPO, especially in the case of high safety probability $1-\varepsilon = 95\%$ (orange for TQPO, purple for QCPO). Table 1 (R Columns) also demonstrates that TQPO outperforms QCPO with higher return. Notably, in SimpleEnv and DynamicEnv (Fig. 3 Row 1, Column 1&2), TQPO with a higher safety level $1-\varepsilon = 95\%$ (orange) may even outperform QCPO with a lower safety level $1-\varepsilon = 90\%$ (purple). The higher return of TQPO not only proves the effectiveness of the quantile estimation method but also demonstrates the advantage of the tilted quantile gradient update. Fig. 4 indicates that the tilted term compensates the asymmetric distributional density of the quantile, ensuring the tilted quantile symmetrically distributed around the threshold. Therefore the decrease of λ is boosted, facilitating its recovery from overshooting and avoiding the over-conservatism of the policy, which leads to higher return for TQPO within the same number of training epochs compared to QCPO.

Furthermore, Directly estimating the quantile through sampling benefits TQPO with greater time efficiency compared to QCPO, which necessitates additional time for distribution fitting and quantile approximation. As shown in Table 1 (T Columns), the average training time for TQPO is approximately 10% shorter than that of QCPO. This indicates that TQPO not only surpasses QCPO in performance but also requires less time for training.

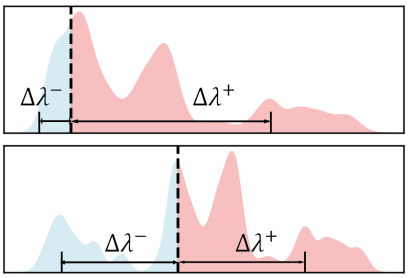


Figure 4: Distributions of quantile $q_{1-\varepsilon}$ w.o. (top) and w. (bottom) tilted term. The black vertical dashed line is the threshold d, $\Delta\lambda^+$ is the increase of λ when $q_{1-\varepsilon} \geq d$, $\Delta\lambda^-$ represent the decrease of λ when $q_{1-\varepsilon} < d$.

Algorithms	SimpleEnv 95%		DynamicEnv 95%	
	R	Pr	R	Pr
QCPO	2.1±0.2	97%±1%	3.5±0.1	97%±0%
QCPO (tilt)	4.7±0.2	**95%±1%**	8.0±0.3	**95%±1%**
TQPO (w/o tilt)	10.3±0.3	96%±1%	5.26±0.2	**95%±1%**
TQPO (fixed tilt)	13.9±0.8	92%±2%	11.9±0.2	92%±1%
TQPO	**14.2±0.3**	**95%±1%**	**12.5±0.2**	94%±1%

Table 2: Ablation study on SimpleEnv 95% and DynamicEnv 95%. R: Average episode return, Pr: Safety probability

Ablation Study

Ablation studies are conducted on two tasks with three variants: QCPO with tilted update, TQPO w/o tilted update and TQPO with fixed tilted rates $\eta_+ = 0.2, \eta_- = 0.8$, as shown in Table 2. First, TQPO(w/o tilt) have higher return than QCPO, validating the effectiveness of the quantile gradient estimation. Second, QCPO(tilt) outperforms QCPO, while TQPO outperforms TQPO(w/o tilt) and TQPO(fixed tilt), indicating the benefit of proposed tilted update. The naive tilted method TQPO(fixed tilt) alleviates early overshooting of λ, but leads to its undershooting later and a biased safety probability 92% eventually. Our tilted update calculates η each epoch to update λ adaptively, results in better safety probability 95%.

Conclusion

In this paper, we have developed a novel quantile-constrained RL model named Tilted Quantile Policy Optimization. This model applies sampling-based quantile gradient estimation for quantile constraints, and a tilted quantile gradient update strategy for higher return. We provide theoretical proofs of the convergence of this TQPO model, which converges to the optimal solution under certain conditions. Experiments on three classic safe RL tasks demonstrate the effectiveness of the proposed TQPO model, which satisfies all the quantile constraints and achieves higher return than the state-of-the-art benchmarks. Our future work will focus on extending this model to multi-agent RL applications and considering the application in real-world systems.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *International conference on machine learning*, 22–31. PMLR.

Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2018. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Altman, E. 2021. *Constrained Markov decision processes*. Routledge.

Borkar, V. S. 1997. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5): 291–294.

Borkar, V. S. 2008. *Stochastic approximation: a dynamical systems viewpoint*, volume 9. Springer.

Carr, S.; Jansen, N.; Junges, S.; and Topcu, U. 2023. Safe reinforcement learning via shielding under partial observability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 14748–14756.

Chen, W.; Subramanian, D.; and Paternain, S. 2024. Probabilistic constraint for safety-critical reinforcement learning. *IEEE Transactions on Automatic Control*.

Cheng, R.; Orosz, G.; Murray, R. M.; and Burdick, J. W. 2019. 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, 3387–3395.

Chow, Y.; Ghavamzadeh, M.; Janson, L.; and Pavone, M. 2018. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167): 1–51.

Gattami, A.; Bai, Q.; and Aggarwal, V. 2021. Reinforcement learning for constrained markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, 2656–2664. PMLR.

Glynn, P. W.; Peng, Y.; Fu, M. C.; and Hu, J.-Q. 2021. Computing sensitivities for distortion risk measures. *INFORMS Journal on Computing*, 33(4): 1520–1532.

Hong, L. J.; and Liu, G. 2009. Simulating sensitivities of conditional value at risk. *Management Science*, 55(2): 281–293.

Jiang, G.; and Fu, M. C. 2015. On estimating quantile sensitivities via infinitesimal perturbation analysis. *Operations Research*, 63(2): 435–441.

Jiang, J.; Peng, Y.; and Hu, J. 2022. Quantile-based policy optimization for reinforcement learning. In *2022 Winter Simulation Conference (WSC)*, 2712–2723. IEEE.

Jung, W.; Cho, M.; Park, J.; and Sung, Y. 2022. Quantile constrained reinforcement learning: A reinforcement learning framework constraining outage probability. *Advances in Neural Information Processing Systems*, 35: 6437–6449.

Liang, Q.; Que, F.; and Modiano, E. 2018. Accelerated primal-dual policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1802.06480*.

Liu, Y.; Ding, J.; and Liu, X. 2020. Ipo: Interior-point policy optimization under constraints. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 4940–4947.

Marvi, Z.; and Kiumarsi, B. 2021. Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control*, 31(6): 1923–1940.

Paternain, S.; Chamon, L.; Calvo-Fullana, M.; and Ribeiro, A. 2019. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32.

Peng, B.; Duan, J.; Chen, J.; Li, S. E.; Xie, G.; Zhang, C.; Guan, Y.; Mu, Y.; and Sun, E. 2022. Model-based chance-constrained reinforcement learning via separated proportional-integral lagrangian. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1): 466–478.

Pfrommer, S.; Gautam, T.; Zhou, A.; and Sojoudi, S. 2022. Safe reinforcement learning with chance-constrained model predictive control. In *Learning for Dynamics and Control Conference*, 291–303. PMLR.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking Safe Exploration in Deep Reinforcement Learning.

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

Steinwart, I.; and Christmann, A. 2011. Estimating conditional quantiles with the help of the pinball loss.

Stooke, A.; and Abbeel, P. 2019. rlpyt: A research code base for deep reinforcement learning in pytorch. *arXiv preprint arXiv:1909.01500*.

Tessler, C.; Mankowitz, D. J.; and Mannor, S. 2018. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*.

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.

Yang, T.-Y.; Rosca, J.; Narasimhan, K.; and Ramadge, P. J. 2020. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*.