Supplementary Material for Trust Region-Based Safe Distributional Reinforcement Learning for Multiple Constraints

Anonymous Author(s)
Affiliation
Address
email

In this supplementary material, we present proofs of theorems in the main text, detailed explanations of the experiments, and additional experimental results. First, Section A shows algorithm details, including proofs of theorems, toy examples, and pseudocode. In particular, Section A.6-8 present additional differences from the existing trust region-based safe RL algorithms. Section B describes the detail of the experimental settings. Section C shows experimental results, including additional ablation experiments. Then, the proposed algorithm is compared with the state-of-the-art traditional RL algorithm in Section D. The traditional RL algorithm is trained with various sets of reward weights, but SDAC shows the highest reward sums among them. These experiments exhibit the challenge of reward tunning when using traditional RL algorithms to perform the locomotion tasks. Finally, Section E presents the computational analysis of the gradient integration method. The table of contents is listed in Table 1.

Table 1: Table of Contents.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

A Algorithm Details

A.1 Proof of Theorem 3.1

We denote the policy parameter space as $\Psi \subseteq \mathbb{R}^d$, the parameter at the tth iteration as $\psi_t \in \Psi$, the Hessian matrix as $H(\psi_t) = \nabla^2_\psi D_{\mathrm{KL}}(\pi_{\psi_t}||\pi_\psi)|_{\psi=\psi_t}$, and the kth cost surrogate as $F_k(\psi_t) = F_k^{\mu,\pi}(\pi_{\psi_t};\alpha)$. As we focus on the tth iteration, the following notations are used for brevity: $H = H(\psi_t)$ and $g_k = \nabla F_k(\psi_t)$. The proposed gradient integration at tth iteration is defined as the following quadratic program (QP):

$$g_t = \underset{g}{\operatorname{argmin}} \frac{1}{2} g^T H g \quad \text{s.t. } g_k^T g + c_k \leq 0 \text{ for } \forall k, \tag{1}$$

where $c_k = \min(\sqrt{2\epsilon g_k^T H^{-1} g_k}, F_k(\pi_\psi;\alpha) - d_k + \zeta)$. In the remainder of this section, we introduce the assumptions and new definitions, discuss the existence of a solution (1), show the convergence to the feasibility condition for varying step size cases, and provide the proof of Theorem 3.1.

Assumption. 1) Each F_k is differentiable and convex, **2)** ∇F_k is L-Lipschitz continuous, **3)** all eigenvalues of the Hessian matrix $H(\psi)$ are equal or greater than $R \in \mathbb{R}_{>0}$ for $\forall \psi \in \Psi$, and **4)** $\{\psi | F_k(\psi) + \zeta < d_k \text{ for } \forall k\} \neq \emptyset$.

Definition. Using the Cholesky decomposition, the Hessian matrix can be expressed as $H = B \cdot B^T$ where B is a lower triangular matrix. By introducing new terms, $\bar{g}_k := B^{-1} g_k$ and $b_t := B^T g_t$, the following is satisfied: $g_k^T H^{-1} g_k = ||\bar{g}_k||_2^2$. Additionally, we define the in-boundary and out-boundary sets as:

$$\mathrm{IB}_k := \left\{ \psi | F_k(\psi) - d_k + \zeta \leq \sqrt{2\epsilon \nabla F_k(\psi)^T H^{-1}(\psi) \nabla F_k(\psi)} \right\},$$

$$\mathrm{OB}_k := \left\{ \psi | F_k(\psi) - d_k + \zeta \geq \sqrt{2\epsilon \nabla F_k(\psi)^T H^{-1}(\psi) \nabla F_k(\psi)} \right\}.$$

The minimum of $||\bar{g}_k||$ in OB_k is denoted as m_k, and the maximum of $||\bar{g}_k||$ in IB_k is denoted as M_k. Also, $\min_k m_k$ and $\max_k M_k$ are denoted as m and M, respectively, and we can say that m is positive.

Lemma A.1. *For all k, the minimum value of m_k is positive.*

Proof. Assume that there exist $k \in \{1, ..., K\}$ such that m_k is equal to zero at a policy parameter $\psi^* \in \mathrm{OB}_k$, i.e., $||\nabla F_k(\psi^*)|| = 0$. Since F_k is convex, ψ^* is a minimum point of F_k, $\min_\psi F_k(\psi) = F_k(\psi^*) < d_k - \zeta$. However, $F_k(\psi^*) \geq d_k - \zeta$ as $\psi^* \in \mathrm{OB}_k$, so m_k is positive due to the contradiction. Hence, the minimum of m_k is also positive. □

Lemma A.2. *A solution of (1) always exists.*

Proof. There exists a policy parameter $\hat{\psi} \in \{\psi | F_k(\psi) + \zeta < d_k \text{ for } \forall k\}$ due to the assumptions. Let $g = \psi - \psi_t$. Then, the following inequality holds.

$$g_k^T(\psi - \psi_t) + c_k \leq g_k^T(\psi - \psi_t) + F_k(\psi_t) + \zeta - d_k \leq F_k(\psi) + \zeta - d_k. \quad (\because F_k \text{ is convex.})$$
$$\Rightarrow g_k^T(\hat{\psi} - \psi_t) + c_k \leq F_k(\hat{\psi}) + \zeta - d_k < 0 \text{ for } \forall k.$$

Since $\hat{\psi} - \psi_t$ satisfies all constraints of (1), the feasible set is non-empty and convex. Also, H is positive definite, so the QP has a unique solution. □

Lemma A.2 shows the existence of solution of (1). We introduce a new lemma, which shows $||b_t||$ is bounded by $\sqrt{\epsilon}$.

Lemma A.3. *There exists $T \in \mathbb{R}_{>0}$ such that $||b_t|| \leq T\sqrt{\epsilon}$.*

Proof. By solving the dual problem of (1), g_t can be expressed as:

$$g_t = -\sum_{k=1}^{K} \lambda_k H^{-1} g_k \text{ s.t. } \lambda_k = \max\left(\frac{c_k - \sum_{j \neq k} \lambda_j g_j^T H^{-1} g_k}{g_k^T H^{-1} g_k}, 0 \right) \text{ for } \forall k.$$

The following inequality holds for $\forall k$:

$$\lambda_k \leq \max\left(\frac{c_k}{||\bar{g}_k||^2}, 0\right) \leq \max\left(\frac{\sqrt{2\epsilon}||\bar{g}_k||}{||\bar{g}_k||^2}, 0\right) \leq \frac{\sqrt{2\epsilon}}{||\bar{g}_k||}.$$

Using triangular inequality,

$$\begin{aligned}||b_t|| = ||B^T g_t|| = ||\sum_k \lambda_k B^T H^{-1} g_k|| &\leq \sum_k \lambda_k ||B^T H^{-1} g_k|| \\ &\leq \sqrt{2\epsilon}\sum_k \frac{||B^T H^{-1} g_k||}{||\bar{g}_k||} = K\sqrt{2\epsilon}.\end{aligned}$$

Hence, for every constant $T > \sqrt{2}K$, the statement holds. □

Now, we show the convergence of the proposed gradient integration method in the case of varying step sizes.

Lemma A.4. *Suppose the following constants κ_1, κ_2 are given: $0 < \kappa_1 < \frac{\sqrt{2}mR\kappa_2}{LK^2\sqrt{\epsilon}}$ and $0 < \kappa_2 < 1$. If $\sqrt{2\epsilon}M \leq \zeta$ and a policy is updated by $\psi_{t+1} = \psi_t + \beta_t g_t$, where $\kappa_1 \leq \beta_t \leq \frac{2\sqrt{2\epsilon}mR}{L||b_t||^2}\kappa_2$, the policy satisfies $F_k(\psi) \leq d_k$ for $\forall k$ within a finite time.*

Proof. We can reformulate the step size as $\beta_t = \frac{2\sqrt{2\epsilon}mR}{L||b_t||^2}\beta'_t$, where $\frac{L||b_t||^2}{2\sqrt{2\epsilon}mR}\kappa_1 \leq \beta'_t \leq \kappa_2$. Since the eigenvalues of H is equal to or bigger than R and H is symmetric and positive definite, $\frac{1}{R}I - H^{-1}$ is positive semi-definite. Hence, $x^T H^{-1} x \leq \frac{1}{R}||x||^2$ is satisfied. Using this fact, the following inequality holds:

$$\begin{aligned}F_k(\psi_t + \beta_t g_t) - F_k(\psi_t) &\leq \beta_t \nabla F_k(\psi_t)^T g_t + \frac{L}{2}||\beta_t g_t||^2 \quad (\because \nabla F_k \text{ is } L\text{-Lipschitz continuous.}) \\ &= \beta_t g_k^T g_t + \frac{L}{2}\beta_t^2 ||g_t||^2 \\ &= \beta_t g_k^T g_t + \frac{L}{2}\beta_t^2 b_t^T H^{-1} b_t \qquad (\because g_t = B^{-T} b_t) \\ &\leq -\beta_t c_k + \frac{L}{2R}\beta_t^2 ||b_t||^2. \qquad (\because g_k^T g_t + c_k \leq 0)\end{aligned}$$

Now, we will show that ψ enters IB_k in a finite time for $\forall \psi \in \text{OB}_k$ and that the kth constraint is satisfied for $\forall \psi \in \text{IB}_k$. Thus, we divide into two cases, **1)** $\psi_t \in \text{OB}_k$ and **2)** $\psi_t \in \text{IB}_k$. For the first case, $c_k = \sqrt{2\epsilon}||\bar{g}_k||$, so the following inequality holds:

$$\begin{aligned}F_k(\psi_t + \beta_t g_t) - F_k(\psi_t) &\leq \beta_t \left(-\sqrt{2\epsilon}||\bar{g}_k|| + \frac{L}{2R}\beta_t ||b_t||^2\right) \\ &\leq \beta_t \sqrt{2\epsilon}\left(-||\bar{g}_k|| + m\beta'_t\right) \\ &\leq \beta_t \sqrt{2\epsilon} m(\beta'_t - 1) \leq \kappa_1(\kappa_2 - 1)\sqrt{2\epsilon}m < 0.\end{aligned} \tag{2}$$

The value of F_k decreases strictly with each update step according to (2). Hence, ψ_t can reach IB_k by repeatedly updating the policy. We now check whether the constraint is satisfied for the second case. For the second case, the following inequality holds by applying $c_k = F_k(\psi_t) - d_k + \zeta$:

$$\begin{aligned}&F_k(\psi_t + \beta_t g_t) - F_k(\psi_t) \leq \beta_t d_k - \beta_t F_k(\psi_t) - \beta_t \zeta + \frac{L}{2R}\beta_t^2 ||b_t||^2 \\ \Rightarrow &F_k(\psi_t + \beta_t g_t) - d_k \leq (1 - \beta_t)(F_k(\psi_t) - d_k) + \beta_t(-\zeta + \sqrt{2\epsilon}m\beta'_t).\end{aligned}$$

Since $\psi_t \in \text{IB}_k$,

$$F_k(\psi_t) - d_k \leq \sqrt{2\epsilon}||\bar{g}_k|| - \zeta \leq \sqrt{2\epsilon}M - \zeta \leq 0.$$

Since $m \leq M$ and $\beta'_t < 1$,

$$-\zeta + \sqrt{2\epsilon}m\beta'_t < -\zeta + \sqrt{2\epsilon}M \leq 0.$$

Hence, $F_k(\psi_t + \beta_t g_t) \leq d_k$, which means that the kth constraint is satisfied if $\psi_t \in \text{IB}_k$. As ψ_t reaches IB_k for $\forall k$ within a finite time according to (2), the policy can satisfy all constraints within a finite time. □

Lemma A.4 shows the convergence to the feasibility condition in the case of varying step sizes. We finally show the proof of Theorem 3.1, which can be considered a special case of varying step sizes.

Theorem 3.1. *Assume that the cost surrogates are differentiable and convex, gradients of the surrogates are L-Lipschitz continuous, eigenvalues of the Hessian are equal or greater than a positive value $R \in \mathbb{R}_{>0}$, and $\{\psi | F_k(\pi_\psi; \alpha) + \zeta < d_k,\ \forall k\} \neq \emptyset$. Then, there exists $E \in \mathbb{R}_{>0}$ such that if $0 < \epsilon \leq E$ and a policy is updated by the proposed gradient integration method, all constraints are satisfied within finite time steps.*

Proof. The proposed step size is $\beta_t = \min(1, \sqrt{2\epsilon}/||b_t||)$, and the sufficient conditions that guarantee the convergence according to Lemma A.4 are followings:

$$\sqrt{2\epsilon}M \leq \zeta, \text{ and } \kappa_1 \leq \beta_t \leq \frac{2\sqrt{2\epsilon}mR}{L||b_t||^2}\kappa_2 \text{ for } \exists \kappa_1, \kappa_2.$$

From the first condition, $\epsilon \leq \zeta^2/(2M^2)$. To satisfy the second condition, the proposed step size β_t should satisfy the followings:

$$\beta_t \leq \frac{\sqrt{2\epsilon}}{||b_t||} \leq \frac{2\sqrt{2\epsilon}mR}{L||b_t||^2}\kappa_2 \ \Leftrightarrow\ ||b_t|| \leq \frac{2mR}{L}\kappa_2.$$

If $\epsilon \leq 2((mR\kappa_2)/(LK))^2$, the following inequality holds:

$$\sqrt{2\epsilon} \leq \frac{2mR}{LK}\kappa_2 \ \Rightarrow\ ||b_t|| \leq K\sqrt{2\epsilon} \leq \frac{2mR}{L}\kappa_2. \quad (\because \text{ Lemma A.3.})$$

If $1 \leq \sqrt{2\epsilon}/||b_t||^2$, it is obvious that $\kappa_1 < 1 = \beta_t$. If $1 > \sqrt{2\epsilon}/||b_t||$, we can get the followings by setting $\kappa_1 \leq 1/K$:

$$\kappa_1 \leq \frac{1}{K} \leq \frac{\sqrt{2\epsilon}}{||b_t||} = \beta_t. \tag{3}$$

Hence, if $\epsilon \leq E = \frac{1}{2}\min(\frac{\zeta^2}{2M^2}, 2(\frac{mR\kappa_2}{LK})^2)$, $0 < \kappa_1 < \min(\frac{\sqrt{2}mR\kappa_2}{LK^2\sqrt{\epsilon}}, \frac{1}{K})$, and $0 < \kappa_2 < 1$, the sufficient conditions are satisfied. □

A.2 Toy Example for Gradient Integration Method

The problem of the toy example in Figure 1 in the main paper is defined as:

$$\underset{x_1,x_2}{\text{minimize}} \sqrt{(\sqrt{3}x_1 + x_2 + 2)^2 + 4(x_1 - \sqrt{3}x_2 + 4)^2} \quad \text{s.t. } x_1 \geq 0,\ x_1 - 2x_2 \leq 0, \tag{4}$$

where there are two linear constraints. The initial points for the naive and gradient integration methods are $x_1 = -2.5$ and $x_2 = -3.0$, which do not satisfied the two constraints. We use the Hessian matrix for the trust region as identity matrix and the trust region size as 0.5 in both methods. The naive method minimizes the constraints in order from the first to the second constraint.

A.3 Proof of Theorem 3.2

In this section, we show that a sequence, $Z_{k+1} = \mathcal{T}_\lambda^{\mu,\pi} Z_k$, converges to the Z_R^π. First, we rewrite the operator $\mathcal{T}_\lambda^{\mu,\pi}$ for random variables to an operator for distributions and show that the operator is contractive. Finally, we show that Z_R^π is the unique fixed point.

Before starting the proof, we introduce useful notions and distance metrics. As the return $Z_R^\pi(s,a)$ is a random variable, we define the distribution of $Z_R^\pi(s,a)$ as $\nu_R^\pi(s,a)$. Let η be the distribution of a random variable X. Then, we can express the distribution of affine transformation of random variable, $aX + b$, using the *pushforward* operator, which is defined by Rowland et al. [2018], as $(f_{a,b})_\#(\eta)$. To measure a distance between two distributions, Bellemare et al. [2023] has defined the distance l_p as follows:

$$l_p(\eta_1, \eta_2) := \left(\int_{\mathbb{R}} |F_{\eta_1}(x) - F_{\eta_2}(x)|^p \, dx \right)^{1/p}, \tag{5}$$

where $F_\eta(x)$ is the cumulative distribution function. This distance is $1/p$-homogeneous, regular, and p-convex (see Section 4 of Bellemare et al. [2023] for more details). For functions that map state-action pairs to distributions, a distance can be defined as [Bellemare et al., 2023]: $\bar{l}_p(\nu_1, \nu_2) := \sup_{(s,a)\in S\times A} l_p(\nu_1(s,a), \nu_2(s,a))$. Then, we can rewrite the operator $\mathcal{T}_\lambda^{\mu,\pi}$ for random variables in (10) as an operator for distributions as below.

$$\begin{aligned}\mathcal{T}_\lambda^{\mu,\pi}\nu(s,a) &:= \frac{1-\lambda}{\mathcal{N}} \sum_{i=0}^{\infty} \lambda^i \\ &\times \mathbb{E}_\mu \left[\left(\prod_{j=1}^{i} \eta(s_j, a_j) \right) \mathbb{E}_{a'\sim\pi(\cdot|s_{i+1})} \left[(f_{\gamma^{i+1}, \sum_{t=0}^i \gamma^t r_t})_\#(\nu(s_{i+1}, a')) \right] \middle| s_0 = s, a_0 = a \right],\end{aligned} \tag{6}$$

where $\eta(s,a) = \frac{\pi(a|s)}{\mu(a|s)}$ and $\mathcal{N}$ is a normalization factor. Since the random variable $Z(s,a)$ and the distribution $\nu(s,a)$ is equivalent, the operators in (10) and (6) are also equivalent. Hence, we are going to show the proof of Theorem 3.2 using (6) instead of (10). We first show that the operator $\mathcal{T}_\lambda^{\mu,\pi}$ has a contraction property.

Lemma A.5. *Under the distance $\bar{l}_p$ and the assumption that the state, action, and reward spaces are finite, $\mathcal{T}_\lambda^{\mu,\pi}$ is $\gamma^{1/p}$-contractive.*

Proof. First, the operator can be rewritten using summation as follows.

$$\begin{aligned}\mathcal{T}_\lambda^{\mu,\pi}\nu(s,a) &= \frac{1-\lambda}{\mathcal{N}} \sum_{i=0}^{\infty} \lambda^i \sum_{a'\in A} \sum_{(s_0,a_0,r_0,\ldots,s_{i+1})} \Pr_\mu(\underbrace{s_0, a_0, r_0, \ldots, s_{i+1}}_{=:\tau}) \left(\prod_{j=1}^{i} \eta(s_j, a_j) \right) \\ &\qquad \times \pi(a'|s_{i+1}) (f_{\gamma^{i+1}, \sum_{t=0}^i \gamma^t r_t})_\#(\nu(s_{i+1}, a')) \\ &= \frac{1-\lambda}{\mathcal{N}} \sum_{i=0}^{\infty} \lambda^i \sum_{a'\in A} \sum_{\tau} \Pr_\mu(\tau) \left(\prod_{j=1}^{i} \eta(s_j, a_j) \right) \pi(a'|s_{i+1}) \sum_{s'\in S} \mathbf{1}_{s'=s_{i+1}} \\ &\qquad \times \sum_{r'_{0:i}} \left(\prod_{k=0}^{i} \mathbf{1}_{r'_k = r_k} \right) (f_{\gamma^{i+1}, \sum_{t=0}^i \gamma^t r'_t})_\#(\nu(s', a')) \\ &= \frac{1-\lambda}{\mathcal{N}} \sum_{i=0}^{\infty} \lambda^i \sum_{a'\in A} \sum_{s'\in S} \sum_{r'_{0:i}} (f_{\gamma^{i+1}, \sum_{t=0}^i \gamma^t r'_t})_\#(\nu(s', a')) \\ &\qquad \times \underbrace{\mathbb{E}_\mu \left[\left(\prod_{j=1}^{i} \eta(s_j, a_j) \right) \pi(a'|s_{i+1}) \mathbf{1}_{s'=s_{i+1}} \left(\prod_{k=0}^{i} \mathbf{1}_{r'_k = r_k} \right) \right]}_{=:w_{s',a',r'_{0:i}}} \\ &= \frac{1-\lambda}{\mathcal{N}} \sum_{i=0}^{\infty} \sum_{s'\in S} \sum_{a'\in A} \sum_{r'_{0:i}} \lambda^i w_{s',a',r'_{0:i}} (f_{\gamma^{i+1}, \sum_{t=0}^i \gamma^t r'_t})_\#(\nu(s', a')).\end{aligned} \tag{7}$$

Since the sum of weights of distributions should be one, we can find the normalization factor $\mathcal{N} = (1-\lambda)\sum_{i=0}^{\infty}\sum_{s\in S}\sum_{a\in A}\sum_{r_{0:i}}\lambda^i w_{s,a,r_{0:i}}$. Then, the following inequality can be derived using the homogeneity, regularity, and convexity of l_p:

$$
\begin{aligned}
&l_p^p(\mathcal{T}_\lambda^{\mu,\pi}\nu_1(s,a), \mathcal{T}_\lambda^{\mu,\pi}\nu_2(s,a)) \\
&= l_p^p\left(\frac{1-\lambda}{\mathcal{N}}\sum_{i=0}^{\infty}\sum_{s\in S}\sum_{a\in A}\sum_{r_{0:i}}\lambda^i w_{s,a,r_{0:i}}(f_{\gamma^{i+1},\sum_{t=0}^{i}\gamma^t r_t})_{\#}(\nu_1(s,a)),\right. \\
&\qquad\left.\frac{1-\lambda}{\mathcal{N}}\sum_{i=0}^{\infty}\sum_{s\in S}\sum_{a\in A}\sum_{r_{0:i}}\lambda^i w_{s,a,r_{0:i}}(f_{\gamma^{i+1},\sum_{t=0}^{i}\gamma^t r_t})_{\#}(\nu_2(s,a))\right) \\
&\le \sum_{i=0}^{\infty}\sum_{s\in S}\sum_{a\in A}\sum_{r_{0:i}}\frac{(1-\lambda)\lambda^i w_{s,a,r_{0:i}}}{\mathcal{N}} l_p^p\left((f_{\gamma^{i+1},\sum_{t=0}^{i}\gamma^t r_t})_{\#}(\nu_1(s,a)),\right. \\
&\qquad\left.(f_{\gamma^{i+1},\sum_{t=0}^{i}\gamma^t r_t})_{\#}(\nu_2(s,a))\right) \qquad (8)\\
&\le \sum_{i=0}^{\infty}\sum_{s\in S}\sum_{a\in A}\sum_{r_{0:i}}\frac{(1-\lambda)\lambda^i w_{s,a,r_{0:i}}}{\mathcal{N}} l_p^p\left((f_{\gamma^{i+1},0})_{\#}(\nu_1(s,a)), (f_{\gamma^{i+1},0})_{\#}(\nu_2(s,a))\right) \\
&= \sum_{i=0}^{\infty}\sum_{s\in S}\sum_{a\in A}\sum_{r_{0:i}}\frac{(1-\lambda)\lambda^i w_{s,a,r_{0:i}}}{\mathcal{N}}\gamma^{i+1} l_p^p\left(\nu_1(s,a),\nu_2(s,a)\right) \\
&\le \sum_{i=0}^{\infty}\sum_{s\in S}\sum_{a\in A}\sum_{r_{0:i}}\frac{(1-\lambda)\lambda^i w_{s,a,r_{0:i}}}{\mathcal{N}}\gamma^{i+1}\left(\bar{l}_p\left(\nu_1,\nu_2\right)\right)^p \\
&\le \gamma\left(\bar{l}_p\left(\nu_1,\nu_2\right)\right)^p.
\end{aligned}
$$

Therefore, $\bar{l}_p\left(\mathcal{T}_\lambda^{\mu,\pi}\nu_1, \mathcal{T}_\lambda^{\mu,\pi}\nu_2\right) \le \gamma^{1/p}\bar{l}_p\left(\nu_1,\nu_2\right)$. □

By the Banach's fixed point theorem, the operator $\mathcal{T}_\lambda^{\mu,\pi}$ has a unique fixed distribution. We now show that the fixed distribution is ν_R^π.

Lemma A.6. *The fixed distribution of the operator $\mathcal{T}_\lambda^{\mu,\pi}$ is ν_R^π.*

Proof. From the definition of Z_R^π, the following equality holds [Rowland et al., 2018]: $\nu_R^\pi(s,a) = \mathbb{E}_\pi\left[(f_{\gamma,r})_{\#}(\nu_R^\pi(s',a'))\right]$. Then, it can be shown that ν_R^π is the fixed distribution by applying the operator $\mathcal{T}_\lambda^{\mu,\pi}$ to ν_R^π:

$$
\begin{aligned}
&\mathcal{T}_\lambda^{\mu,\pi}\nu_R^\pi(s,a) = \frac{1-\lambda}{\mathcal{N}}\sum_{i=0}^{\infty}\lambda^i \\
&\times \mathbb{E}_\mu\left[\left(\prod_{j=1}^{i}\eta(s_j,a_j)\right)\mathbb{E}_{a'\sim\pi(\cdot|s_{i+1})}\left[(f_{\gamma^{i+1},\sum_{t=0}^{i}\gamma^t r_t})_{\#}(\nu_R^\pi(s_{i+1},a'))\right]\Big| s_0=s, a_0=a\right] \qquad (9)\\
&= \frac{1-\lambda}{\mathcal{N}}\sum_{i=0}^{\infty}\lambda^i\mathbb{E}_\pi\left[(f_{\gamma^{i+1},\sum_{t=0}^{i}\gamma^t r_t})_{\#}(\nu_R^\pi(s_{i+1},a_{i+1}))\Big| s_0=s, a_0=a\right] \\
&= \frac{1-\lambda}{\mathcal{N}}\sum_{i=0}^{\infty}\lambda^i\nu_R^\pi(s,a) = \nu_R^\pi(s,a).
\end{aligned}
$$

□

Theorem 3.2. *Let define a distributional operator $\mathcal{T}_\lambda^{\mu,\pi}$, whose probability density function is:*

$$
\begin{aligned}
&\Pr(\mathcal{T}_\lambda^{\mu,\pi}Z(s,a)=z) \propto \\
&\sum_{i=0}^{\infty}\mathbb{E}_\mu\left[\lambda^i\prod_{j=1}^{i}\frac{\pi(a_j|s_j)}{\mu(a_j|s_j)}\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s_{i+1})}\left[\Pr\left(\sum_{t=0}^{i}\gamma^t R_t+\gamma^{i+1}Z(s_{i+1},a')=z\right)\right]\Big| s_0=s, a_0=a\right].
\end{aligned} \qquad (10)
$$

Then, a sequence, $Z_{k+1}(s,a) = \mathcal{T}_\lambda^{\mu,\pi}Z_k(s,a)\ \forall(s,a)$, converges to Z_R^π.

Proof. The operator $\mathcal{T}_\lambda^{\mu,\pi}$ is $\gamma^{1/p}$-contractive under the distance $\bar{l}_p$ according to Lemma A.5. Also, the fixed distribution of the operator is ν_R^π, which is equivalent to Z_R^π, according to Lemma A.6. By the Banach's fixed point theorem, the sequence, $Z_{k+1}(s,a) = \mathcal{T}_\lambda^{\mu,\pi} Z_k(s,a)\ \forall(s,a)$, converges to the fixed distribution of the operator, Z_R^π. □

A.4 Pseudocode of TD(λ) Target Distribution

We provide the pseudocode for calculating TD(λ) target distribution for the reward critic in Algorithm 1. The target distribution for the cost critics can also be obtained by simply replacing the reward part with the cost.

Algorithm 1 TD(λ) Target Distribution

Input: Policy network π_ψ, critic network Z_θ^π, and trajectory $\{(s_t, a_t, \mu(a_t|s_t), r_t, d_t, s_{t+1})\}_{t=1}^T$.
Sample an action $a'_{T+1} \sim \pi_\psi(s_{T+1})$ and get $\hat{Z}_T^{\text{tot}} = r_T + (1-d_T)\gamma Z_\theta^\pi(s_{T+1}, a'_{T+1})$.
Initialize the total weight $w_{\text{tot}} = \lambda$.
for $t = T$ **to** 1 **do**
  Sample an action $a'_{t+1} \sim \pi_\psi(s_{t+1})$ and get $\hat{Z}_t^{(1)} = r_t + (1-d_t)\gamma Z_\theta^\pi(s_{t+1}, a'_{t+1})$.
  Set the current weight $w = 1 - \lambda$.
  Combine the two targets, $(\hat{Z}_t^{(1)}, w)$ and $(\hat{Z}_t^{(\text{tot})}, w_{\text{tot}})$, and sort the combined target according to the positions of atoms.
  Build the CDF of the combined target by accumulating the weights at each atom.
  Project the combined target into a quantile distribution with M' atoms, which is $\hat{Z}_t^{(\text{proj})}$, using the CDF (find the atom positions corresponding to each quantile).
  Update $\hat{Z}_{t-1}^{(\text{tot})} = r_{t-1} + (1-d_{t-1})\gamma \hat{Z}_t^{(\text{proj})}$ and $w_{\text{tot}} = \lambda \frac{\pi_\psi(a_t|s_t)}{\mu(a_t|s_t)}(1-d_{t-1})(1-\lambda+w_{\text{tot}})$.
end for
Return $\{\hat{Z}_t^{(\text{proj})}\}_{t=1}^T$.

A.5 Quantitative Analysis on TD(λ) Target Distribution

We experiment with a toy example to measure the bias and variance of the reward estimation according to λ. The toy example has two states, s_1 and s_2; the state distribution is defined as an uniform; the reward function is defined as $r(s_1) \sim \mathcal{N}(-0.005, 0.02)$ and $r(s_2) \sim \mathcal{N}(0.005, 0.03)$. We train parameterized reward distributions by minimizing the quantile regression loss with the TD(λ) target distribution for $\lambda = 0, 0.5, 0.9$, and 1.0. The experimental results are presented in the table below.

Table 2: Experimental results of the toy example.

	5th iteration	10th iteration	15th iteration	20th iteration	25th iteration
$\lambda = 0.0$	4.813 (0.173)	4.024 (0.253)	3.498 (0.085)	3.131 (0.103)	2.835 (0.070)
$\lambda = 0.5$	4.621 (0.185)	3.688 (0.273)	2.925 (0.183)	2.379 (0.134)	2.057 (0.070)
$\lambda = 0.9$	4.141 (0.461)	2.237 (0.402)	1.389 (0.132)	1.058 (0.031)	0.923 (0.019)
$\lambda = 1.0$	2.886 (0.767)	1.733 (0.365)	1.509 (0.514)	1.142 (0.325)	1.109 (0.476)

The values in the table are the mean and standard deviation of the past five values of the Wasserstein distance between the true reward return and the estimated distribution. Looking at the fifth iteration, it is clear that the larger the λ value, the smaller the mean and the higher the standard deviation. At the 25th iteration, the run with $\lambda = 0.9$ has the lowest mean and standard deviation, indicating that training has converged. On the other hand, the run with $\lambda = 1.0$ has the biggest standard deviation, and the mean is greater than when $\lambda = 0.9$, indicating that the significant variance hinders training. In conclusion, we measured bias and variance quantitatively through the toy example, and the results are well aligned with our claim that λ can trade off bias and variance.

A.6 Policy Update Rule

To solve the constrained optimization problem (6) in the main text, we find a policy update direction by linearly approximating the objective and safety constraints and quadratically approximating the trust region constraint, as done by Achiam et al. [2017]. After finding the direction, we update the policy using a line search method. Given the current policy parameter $\psi_t \in \Psi$, the approximated problem can be expressed as follows:

$$x^* = \underset{x \in \Psi}{\operatorname{argmax}}\, g^T x \quad \text{s.t. } \frac{1}{2} x^T H x \leq \epsilon,\ b_k^T x + c_k \leq 0\ \forall k, \tag{11}$$

where $g = \nabla_\psi J^{\mu,\pi}(\pi_\psi)|_{\psi=\psi_t}$, $H = \nabla_\psi^2 D_{\mathrm{KL}}(\pi_{\psi_t}||\pi_\psi)|_{\psi=\psi_t}$, $b_k = \nabla_\psi F_k^{\mu,\pi}(\pi_\psi;\alpha)|_{\psi=\psi_t}$, and $c_k = F_k(\pi_\psi;\alpha) - d_k$. Since (11) is convex, we can use an existing convex optimization solver. However, the search space, which is the policy parameter space Ψ, is excessively large, so we reduce the space by converting (11) to a dual problem as follows:

$$\begin{aligned} g(\lambda,\nu) &= \min_x L(x,\lambda,\nu) = \min_x \{-g^T x + \nu(\frac{1}{2}x^T H x - \epsilon) + \lambda^T(Bx + c)\} \\ &= \frac{-1}{2\nu}\left(\underbrace{g^T H^{-1} g}_{=:q} - 2\underbrace{g^T H^{-1} B^T}_{=:r^T}\lambda + \lambda^T \underbrace{B H^{-1} B^T}_{=:S}\lambda \right) + \lambda^T c - \nu\epsilon \\ &= \frac{-1}{2\nu}(q - 2r^T\lambda + \lambda^T S\lambda) + \lambda^T c - \nu\epsilon, \end{aligned} \tag{12}$$

where $B = (b_1, .., b_K)$, $c = (c_1, ..., c_K)^T$, and $\lambda \in \mathbb{R}^K \geq 0$ and $\nu \in \mathbb{R} \geq 0$ are Lagrange multipliers. Then, the optimal λ and ν can be obtained by a convex optimization solver. After obtaining the optimal values, $(\lambda^*, \nu^*) = \operatorname{argmax}_{(\lambda,\nu)} g(\lambda,\nu)$, the policy update direction x^* are calculated by $\frac{1}{\nu^*} H^{-1}(g - B^T\lambda^*)$. Then, the policy is updated by $\psi_{t+1} = \psi_t + \beta x^*$, where β is a step size, which can be found through a backtracking method (please refer to Section 6.3.2 of Dennis and Schnabel [1996]).

Before using the above policy update rule, we should note that the existing trust-region method with the risk-averse constraint [Kim and Oh, 2022a] and the equations (1, 5, 6) of the main text are slightly different. There are two differences: 1) the objective is augmented with an entropy bonus, and 2) the surrogates are expressed with Q-functions instead of value functions. To use the entropy-regularized objective for the trust region method, it is required to show that the objective is bounded by the KL divergence. We present the existence of bound in Appendix A.7. Next, there is no problem using the Q-functions because it is mathematically equivalent between the original surrogates [Kim and Oh, 2022a] and the new ones expressed with Q-functions defined in (5) of the main text. However, we experimentally show that using the Q-functions in off-policy settings has advantages in Appendix A.8.

A.7 Bound of Entropy-Augmented Objective

In the main text, the objective of the safe RL problem is augmented by entropy regularization as follows:

$$J(\pi) := \mathbb{E}\left[Z_R^\pi(s,a)|s \sim \rho, a \sim \pi(\cdot|s)\right] + \beta\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t H(\pi(\cdot|s_t))|\rho,\pi,P\right]. \tag{13}$$

To use the entropy-regularized objective for the trust region method, it is required to show that the objective is bounded by the KL divergence. To this end, we show that the entropy-regularzied objective in (13) has a bound expressed by the KL divergence in this section. Before showing the boundness, we present a new function and a lemma. A value difference function is defined as follows:

$$\delta^{\pi'}(s) := \mathbb{E}\left[R(s,a,s') + \gamma V^\pi(s') - V^\pi(s) \mid a \sim \pi'(\cdot|s), s' \sim P(\cdot|s,a)\right] = \mathop{\mathbb{E}}_{a\sim\pi'}\left[A^\pi(s,a)\right],$$

where $A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s,a)$.

Lemma A.7. *The maximum of $|\delta^{\pi'}(s) - \delta^\pi(s)|$ is equal or less than $\epsilon_R\sqrt{2D_{\mathrm{KL}}^{\max}(\pi||\pi')}$, where $\epsilon_R = \max_{s,a}|A^\pi(s,a)|$.*

Proof. The value difference can be expressed in a vector form,

$$\delta^{\pi'}(s) - \delta^{\pi}(s) = \sum_a (\pi'(a|s) - \pi(a|s)) A^{\pi}(s,a) = \langle \pi'(\cdot|s) - \pi(\cdot|s), A^{\pi}(s,\cdot) \rangle.$$

Using Hölder's inequality, the following inequality holds:

$$\begin{aligned} |\delta^{\pi'}(s) - \delta^{\pi}(s)| &\leq ||\pi'(\cdot|s) - \pi(\cdot|s)||_1 \cdot ||A^{\pi}(s,\cdot)||_\infty \\ &= 2D_{\mathrm{TV}}(\pi'(\cdot|s)||\pi(\cdot|s)) \max_a A^{\pi}(s,a). \end{aligned}$$

$$\Rightarrow ||\delta^{\pi'} - \delta^{\pi}||_\infty = \max_s |\delta^{\pi'}(s) - \delta^{\pi}(s)| \leq 2\epsilon_R \max_s D_{\mathrm{TV}}(\pi(\cdot|s)||\pi'(\cdot|s)).$$

Using Pinsker's inequality, $||\delta^{\pi'} - \delta^{\pi}||_\infty \leq \epsilon_R \sqrt{2 D_{\mathrm{KL}}^{\max}(\pi||\pi')}$. □

Theorem A.8. *Let us assume that* $\max_s H(\pi(\cdot|s)) < \infty$ *for* $\forall \pi \in \Pi$. *The difference between the objective and surrogate functions is bounded by a term consisting of KL divergence as:*

$$\left| J(\pi') - J^{\mu,\pi}(\pi') \right| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2} \sqrt{D_{\mathrm{KL}}^{\max}(\pi||\pi')} \left(\beta\epsilon_H + \epsilon_R \sqrt{2 D_{\mathrm{KL}}^{\max}(\mu||\pi')} \right), \tag{14}$$

where $\epsilon_H = \max_s |H(\pi'(\cdot|s))|$, $D_{\mathrm{KL}}^{\max}(\pi||\pi') = \max_s D_{\mathrm{KL}}(\pi(\cdot|s)||\pi'(\cdot|s))$, *and the equality holds when* $\pi' = \pi$.

Proof. The surrogate function can be expressed in vector form as follows:

$$J^{\mu,\pi}(\pi') = \langle \rho, V^{\pi} \rangle + \frac{1}{1-\gamma} \left(\langle d^{\mu}, \delta^{\pi'} \rangle + \beta \langle d^{\pi}, H^{\pi'} \rangle \right),$$

where $H^{\pi'}(s) = H(\pi'(\cdot|s))$. The objective function of π' can also be expressed in a vector form using Lemma 1 from Achiam et al. [2017],

$$\begin{aligned} J(\pi') &= \frac{1}{1-\gamma} \mathbb{E}\left[R(s,a,s') + \beta H^{\pi'}(s) \mid s \sim d^{\pi'}, a \sim \pi'(\cdot|s), s' \sim P(\cdot|s,a) \right] \\ &= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s \sim d^{\pi'}} \left[\delta^{\pi'}(s) + \beta H^{\pi'}(s) \right] + \mathop{\mathbb{E}}_{s \sim \rho} \left[V^{\pi}(s) \right] \\ &= \langle \rho, V^{\pi} \rangle + \frac{1}{1-\gamma} \langle d^{\pi'}, \delta^{\pi'} + \beta H^{\pi'} \rangle. \end{aligned}$$

By Lemma 3 from Achiam et al. [2017], $||d^{\pi} - d^{\pi'}||_1 \leq \frac{\gamma}{1-\gamma} \sqrt{2 D_{\mathrm{KL}}^{\max}(\pi||\pi')}$. Then, the following inequality is satisfied:

$$\begin{aligned} &|(1-\gamma)(J^{\mu,\pi}(\pi') - J(\pi'))| \\ &\quad = |\langle d^{\pi'} - d^{\mu}, \delta^{\pi'} \rangle + \beta \langle d^{\pi} - d^{\pi'}, H^{\pi'} \rangle| \\ &\quad \leq |\langle d^{\pi'} - d^{\mu}, \delta^{\pi'} \rangle| + \beta |\langle d^{\pi} - d^{\pi'}, H^{\pi'} \rangle| \\ &\quad = |\langle d^{\pi'} - d^{\mu}, \delta^{\pi'} - \delta^{\pi} \rangle| + \beta |\langle d^{\pi} - d^{\pi'}, H^{\pi'} \rangle| && (\because \delta^{\pi} = 0) \\ &\quad \leq ||d^{\pi'} - d^{\mu}||_1 ||\delta^{\pi'} - \delta^{\pi}||_\infty + \beta ||d^{\pi} - d^{\pi'}||_1 ||H^{\pi'}||_\infty && (\because \text{Hölder's inequality}) \\ &\quad \leq \frac{2\epsilon_R \gamma}{1-\gamma} \sqrt{D_{\mathrm{KL}}^{\max}(\mu||\pi') D_{\mathrm{KL}}^{\max}(\pi||\pi')} + \frac{\beta\gamma\epsilon_H}{1-\gamma} \sqrt{2 D_{\mathrm{KL}}^{\max}(\pi||\pi')} && (\because \text{Lemma A.7}) \\ &\quad = \frac{\gamma}{1-\gamma} \sqrt{D_{\mathrm{KL}}^{\max}(\pi||\pi')} \left(\sqrt{2}\beta\epsilon_H + 2\epsilon_R \sqrt{D_{\mathrm{KL}}^{\max}(\mu||\pi')} \right). \end{aligned}$$

If $\pi' = \pi$, the KL divergence term becomes zero, so equality holds. □

A.8 Comparison of Q-Function and Value Function-Based Surrogates

The original surrogate is defined as follows:

$$J^{\mu,\pi}(\pi') := J(\pi) + \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{d^{\mu},\mu} \left[\frac{\pi'(a|s)}{\mu(a|s)} A^{\pi}(s,a) \right], \tag{15}$$

where $A^{\pi}(s,a) := Q^{\pi}(s,a) - V^{\pi}(s,a)$, and the surrogate is the same as that of OffTRPO [Meng et al., 2022] and OffTRC [Kim and Oh, 2022a]. An entropy-regularized version can be derived as:

$$J^{\mu,\pi}(\pi') = J(\pi) + \frac{1}{1-\gamma} \Big(\beta \mathop{\mathbb{E}}_{d^{\pi}} \left[H(\pi'(\cdot|s)) \right] + \mathop{\mathbb{E}}_{d^{\mu},\mu} \left[\frac{\pi'(a|s)}{\mu(a|s)} A^{\pi}(s,a) \right] \Big). \tag{16}$$

Then, the surrogate expressed by Q-functions in (5) of the main text, called SAC-style version, can be rewritten as:

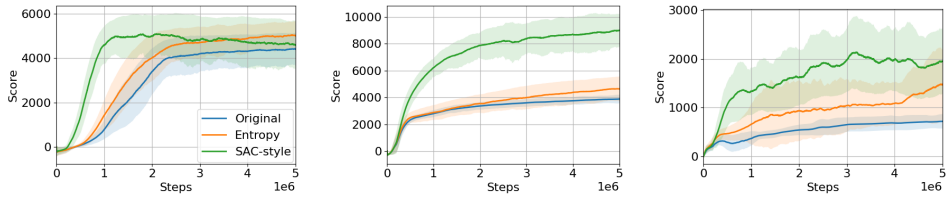
$$J^{\mu,\pi}(\pi') = J(\pi) + \frac{1}{1-\gamma} \Big(\beta \mathop{\mathbb{E}}_{d^{\pi}} \left[H(\pi'(\cdot|s)) \right] + \mathop{\mathbb{E}}_{d^{\mu},\pi'} \left[Q^{\pi}(s,a) \right] \Big). \tag{17}$$

In this section, we evaluate the original, entropy-regularized, and SAC-style versions in the continuous control tasks of the MuJoCo simulators [Todorov et al., 2012]. We use neural networks with two hidden layers with (512, 512) nodes and ReLU for the activation function. The output of a value network is linear, but the input is different; the original and entropy-regularized versions use states, and the SAC-style version uses state-action pairs. The input of a policy network is the state, the output is mean μ and std σ, and actions are squashed into $\tanh(\mu + \epsilon\sigma)$, $\epsilon \sim \mathcal{N}(0,1)$ as in SAC [Haarnoja et al., 2018]. The entropy coefficient β in the entropy-regularized and SAC-style versions are adaptively adjusted to keep the entropy above a threshold (set as $-d$ given $A \subseteq \mathbb{R}^d$). The hyperparameters for all versions are summarized in Table 3.

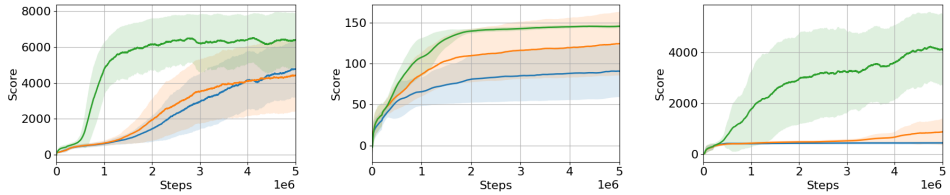
Table 3: Hyperparameters for all versions.

Parameter	Value
Discount factor γ	0.99
Trust region size ϵ	0.001
Length of replay buffer	10^5
Critic learning rate	0.0003
Trace-decay λ	0.97
Initial entropy coefficient β	1.0
β learning rate	0.01

The training curves are presented in Figure 1. All methods are trained with five different random seeds. Although the entropy-regularized version (16) and SAC-style version (17) are mathematically equivalent, it can be observed that the performance of the SAC-style version is superior to the regularized version. It can be inferred that this is due to the variance of importance sampling. In the off-policy setting, the sampling probabilities of the behavioral and current policies can be significantly different, so the variance of the importance ratio is huge. The increased variance prevents estimating the objective accurately, so significant performance degradation can happen. As a result, using the Q-function-based surrogates has an advantage for efficient learning.



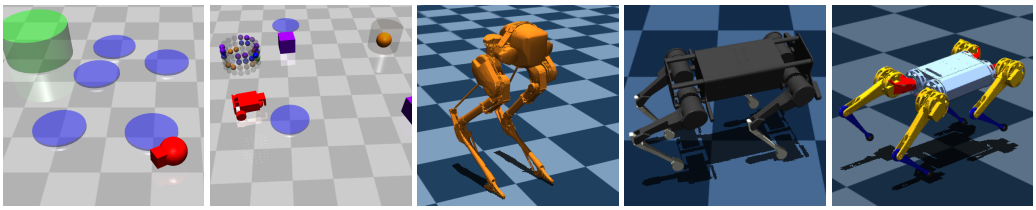
(a) Ant-v3 (b) HalfCheetah-v3 (c) Hopper-v3



(d) Humanoid-v3 (e) Swimmer-v3 (f) Walker2d-v3

Figure 1: MuJoCo training curves.

B Experimental Settings



(a) Point goal. (b) Car button. (c) Cassie. (d) Laikago. (e) Mini-Cheetah.

Figure 2: (a) and (b) are Safety Gym tasks. (c), (d), and (e) are locomotion tasks.

Safety Gym. We use the goal and button tasks with the point and car robots in the Safety Gym environment [Ray et al., 2019], as shown in Figure 2a and 2b. The environmental setting for the goal task is the same as in Kim and Oh [2022b]. Eight hazard regions and one goal are randomly spawned at the beginning of each episode, and a robot gets a reward and cost as follows:

$$\begin{aligned} R(s,a,s') &= -\Delta d_{\text{goal}} + \mathbf{1}_{d_{\text{goal}} \leq 0.3}, \\ C(s,a,s') &= \text{Sigmoid}(10 \cdot (0.2 - d_{\text{hazard}})), \end{aligned} \tag{18}$$

where d_{goal} is the distance to the goal, and d_{hazard} is the minimum distance to hazard regions. If d_{goal} is less than or equal to 0.3, a goal is respawned. The state consists of relative goal position, goal distance, linear and angular velocities, acceleration, and LiDAR values. The action space is two-dimensional, which consists of xy-directional forces for the point and wheel velocities for the car robot.

The environmental settings for the button task are the same as in Liu et al. [2022]. There are five hazard regions, four dynamic obstacles, and four buttons, and all components are fixed throughout the training. The initial position of a robot and an activated button are randomly placed at the beginning of each episode. The reward function is the same as in (18), but the cost is different since there is no dense signal for contacts. We define the cost function for the button task as an indicator function that outputs one if the robot makes contact with an obstacle or an inactive button or enters a hazardous region. We add LiDAR values of buttons and obstacles to the state of the goal task, and actions are the same as the goal task. The length of the episode is 1000 steps without early termination.

Locomotion Tasks. We use three different legged robots, Mini-Cheetah, Laikago, and Cassie, for the locomotion tasks, as shown in Figure 2e, 2d, and 2c. The tasks aim to control robots to follow a velocity command on flat terrain. A velocity command is given by $(v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_z^{\text{cmd}})$, where

$v_x^{\text{cmd}} \sim \mathcal{U}(-1.0, 1.0)$ for Cassie and $\mathcal{U}(-1.0, 2.0)$ otherwise, $v_y^{\text{cmd}} = 0$, and $\omega_z^{\text{cmd}} \sim \mathcal{U}(-0.5, 0.5)$. To lower the task complexity, we set the y-directional linear velocity to zero but can scale to any non-zero value. As in other locomotion studies [Lee et al., 2020, Miki et al., 2022], *central phases* are introduced to produce periodic motion, which are defined as $\phi_i(t) = \phi_{i,0} + f \cdot t$ for $\forall i \in \{1, ..., n_{\text{legs}}\}$, where f is a frequency coefficient and is set to 10, and $\phi_{i,0}$ is an initial phase. Actuators of robots are controlled by PD control towards target positions given by actions. The state consists of velocity command, orientation of the robot frame, linear and angular velocities of the robot, positions and speeds of the actuators, central phases, history of positions and speeds of the actuators (past two steps), and history of actions (past two steps). A foot contact timing ξ can be defined as follows:

$$\xi_i(s) = -1 + 2 \cdot \mathbf{1}_{\sin(\phi_i) \leq 0} \ \ \forall i \in \{1, ..., n_{\text{legs}}\}, \tag{19}$$

where a value of -1 means that the ith foot is on the ground; otherwise, the foot is in the air. For the quadrupedal robots, Mini-Cheetah and Laikago, we use the initial phases as $\phi_0 = \{0, \pi, \pi, 0\}$, which generates trot gaits. For the bipedal robot, Cassie, the initial phases are defined as $\phi_0 = \{0, \pi\}$, which generates walk gaits. Then, the reward and cost functions are defined as follows:

$$\begin{gathered} R(s, a, s') = -0.1 \cdot (||v_{x,y}^{\text{base}} - v_{x,y}^{\text{cmd}}||_2^2 + ||\omega_z^{\text{base}} - \omega_z^{\text{cmd}}||_2^2 + 10^{-3} \cdot R_{\text{power}}), \\ C_1(s, a, s') = \mathbf{1}_{\text{angle} \geq a}, \ C_2(s, a, s') = \mathbf{1}_{\text{height} \leq b}, \ C_3(s, a, s') = \sum_{i=1}^{n_{\text{legs}}} (1 - \xi_i \cdot \hat{\xi}_i)/(2 \cdot n_{\text{legs}}), \end{gathered} \tag{20}$$

where the power consumption $R_{\text{power}} = \sum_i |\tau_i v_i|$, the sum of the torque times the actuator speed, is added to the reward as a regularization term, $v_{x,y}^{\text{base}}$ is the xy-directional linear velocity of the base frame of robots, ω_z^{base} is the z-directional angular velocity of the base frame, and $\hat{\xi} \in \{-1, 1\}^{n_{\text{legs}}}$ is the current feet contact vector. For balancing, the first cost indicates whether the angle between the z-axis vector of the robot base and the world is greater than a threshold ($a = 15^\circ$ for all robots). For standing, the second cost indicates the height of CoM is less than a threshold ($b = 0.3, 0.35, 0.7$ for Mini-Cheetah, Laikago, and Cassie, respectively), and the last cost is to check that the current feet contact vector $\hat{\xi}$ matches the pre-defined timing ξ. The length of the episode is 500 steps. There is no early termination, but if a robot falls to the ground, the state is frozen until the end of the episode.

Hyperparameter Settings. The structure of neural networks consists of two hidden layers with $(512, 512)$ nodes and ReLU activation for all baselines and the proposed method. The input of value networks is state-action pairs, and the output is the positions of atoms. The input of policy networks is the state, the output is mean μ and std σ, and actions are squashed into $\tanh(\mu + \epsilon\sigma)$, $\epsilon \sim \mathcal{N}(0, 1)$. We use a fixed entropy coefficient β. The trust region size ϵ is set to 0.001 for all trust region-based methods. The overall hyperparameters for the proposed method can be summarized in Table 4.

Table 4: Hyperparameter settings for the Safety Gym and locomotion tasks.

Parameter	Safety Gym	Locomotion
Discount factor γ	0.99	0.99
Trust region size ϵ	0.001	0.001
Length of replay buffer	10^5	10^5
Critic learning rate	0.0003	0.0003
Trace-decay λ	0.97	0.97
Entropy coefficient β	0.0	0.001
The number of critic atoms M	25	25
The number of target atoms M'	50	50
Constraint risk level α	0.25, 0.5, and 1.0	1.0
threshold d_k	$0.025/(1-\gamma)$	$[0.025, 0.025, 0.4]/(1-\gamma)$
Slack coefficient ζ	-	$\min_k d_k = 0.025/(1-\gamma)$

Since the range of the cost is $[0, 1]$, the maximum discounted cost sum is $1/(1-\gamma)$. Thus, the threshold is set by target cost rate times $1/(1-\gamma)$. For the locomotion tasks, the third cost in (20) is designed for foot stamping, which is not essential to safety. Hence, we set the threshold to near the maximum (if a robot does not stamp, the cost rate becomes 0.5). In addition, baseline safe RL methods use multiple critic networks for the cost function, such as target [Yang et al., 2021] or square value networks [Kim and Oh, 2022a]. To match the number of network parameters, we use two critics as an ensemble, as in Kuznetsov et al. [2020].

Tips for Hyperparameter Tuning.

- Discount factor γ, Critic learning rate: Since these are commonly used hyperparameters, we do not discuss these.
- Trace-decay λ, Trust region size ϵ: The ablation studies on these hyperparameters are presented in Appendix C.3. From the results, we recommend setting the trace-decay to $0.95 \sim 0.99$ as in other TD(λ)-based methods [Precup et al., 2000]. Also, the results show that the performance is not sensitive to the trust region size. However, if the trust region size is too large, the approximation error increases, so it is better to set it below 0.003.
- Entropy coefficient β: This value is fixed in our experiments, but it can be adjusted automatically as done in SAC [Haarnoja et al., 2018].
- The number of atoms M, M': Although experiments on the number of atoms did not performed, performance is expected to increase as the number of atoms increases, as in other distributional RL methods [Dabney et al., 2018].
- Length of replay buffer: The effect of the length of the replay buffer can be confirmed through the experimental results from an off policy-based safe RL method [Kim and Oh, 2022a]. According to that, the length does not impact performance unless it is too short. We recommend setting it to 10 to 100 times the collected trajectory length.
- Constraint risk level α, threshold d_k: If the cost sum follows a Gaussian distribution, the mean-std constraint is identical to the CVaR constraint. Then, the probability of the worst case can be controlled by adjusting α. For example, if we set $\alpha = 0.125$ and $d = 0.03/(1-\gamma)$, the mean-std constraint enforces the probability that the average cost is less than 0.03 during an episode greater than $95\% = \Phi(\phi(\Phi^{-1}(\alpha))/\alpha)$. Through this meaning, proper α and d_k can be found.
- Slack coefficient ζ: As mentioned at the end of Section 3.1, it is recommended to set this coefficient as large as possible. Since $d_k - \zeta$ should be positive, we recommend setting ζ to $\min_k d_k$.

In conclusion, most hyperparameters are not sensitive, so few need to be optimized. It seems that α and d_k need to be set based on the meaning described above. Additionally, if the approximation error of critics is significant, the trust region size should be set smaller.

C Experimental Results

C.1 Safety Gym

In this section, we present the training curves of the Safety Gym tasks separately according to the risk level of constraints for better readability. Figure 3 shows The training results of the risk-neutral constrained algorithms and risk-averse constrained algorithms with $\alpha = 1.0$. Figures 4 and 5 show the training results of the risk-averse constrained algorithms with $\alpha = 0.25$ and 0.5, respectively.

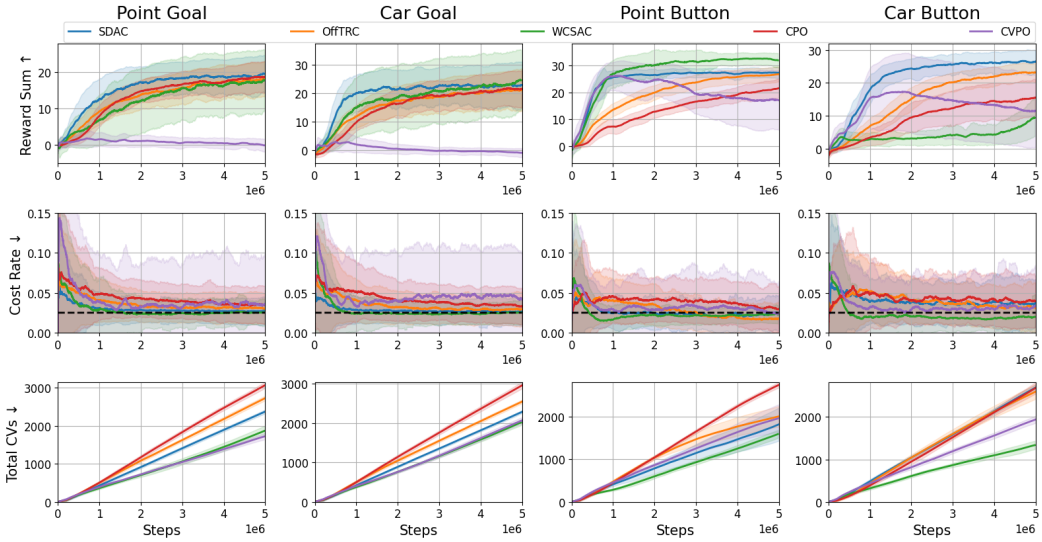


Figure 3: Training curves of risk-neutral constrained algorithms for the Safety Gym tasks. The solid line and shaded area represent the average and std values, respectively. The black dashed lines in the second row indicate thresholds. All methods are trained with five random seeds.

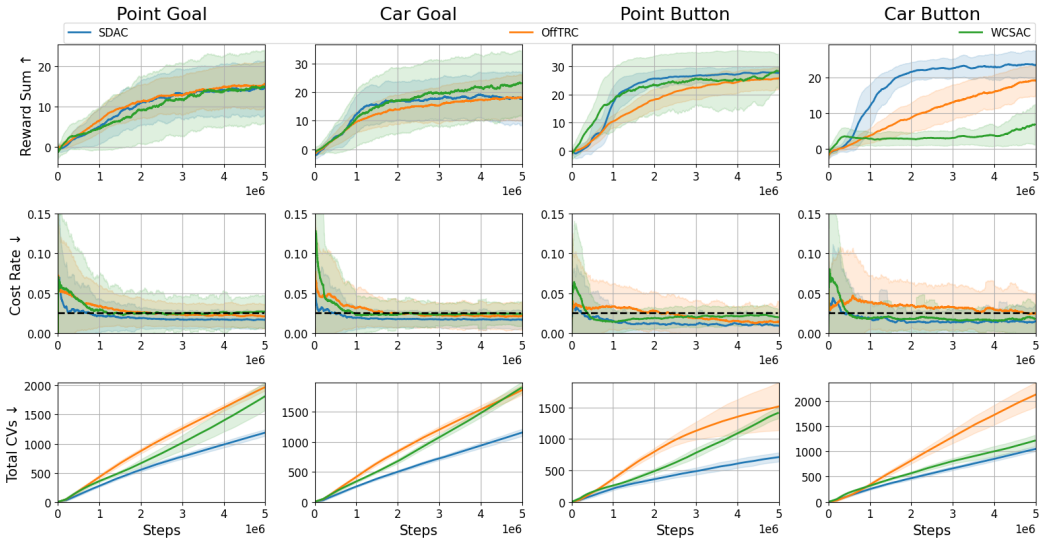


Figure 4: Training curves of risk-averse constrained algorithms with $\alpha = 0.5$ for the Safety Gym.

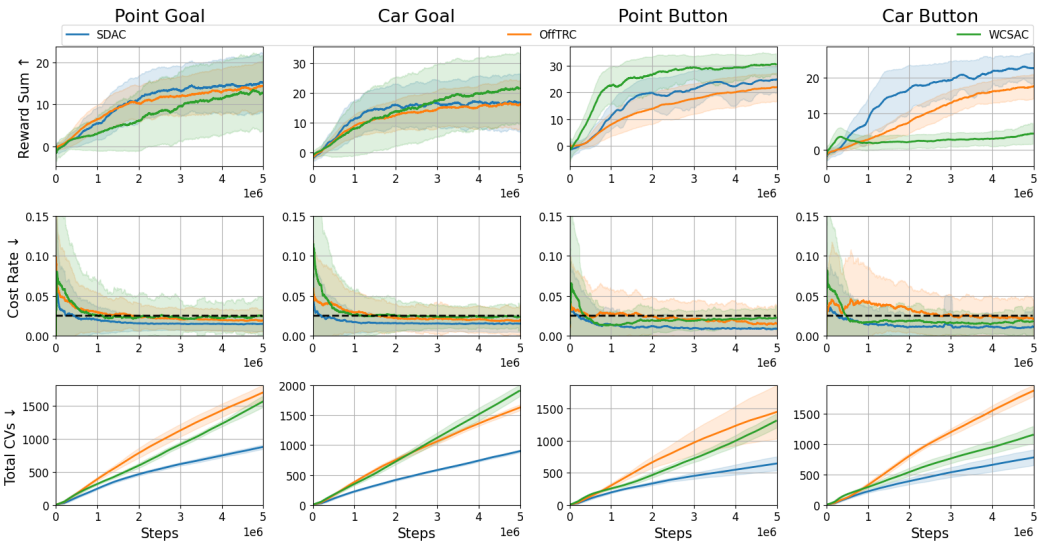


Figure 5: Training curves of risk-averse constrained algorithms with $\alpha = 0.25$ for the Safety Gym.

C.2 Ablation Study on Components of SDAC

There are three main differences between SDAC and the existing trust region-based safe RL algorithm for mean-std constraints [Kim and Oh, 2022a], called OffTRC: 1) feasibility handling methods in multi-constraint settings, 2) the use of distributional critics, and 3) the use of Q-functions instead of advantage functions, as explained in Appendix A.6 and A.8. Since the ablation study for feasibility handling is conducted in Section 5.3, we perform ablation studies for the distributional critic and Q-function in this section. We call SDAC with only distributional critics as *SDAC-Dist* and SDAC with only Q-functions as *SDAC-Q*. If all components are absent, SDAC is identical to OffTRC [Kim and Oh, 2022a]. The variants are trained with the point goal task of the Safety Gym, and the training results are shown in Figure 6. SDAC-Q lowers the cost rate quickly but shows the lowest score. SDAC-Dist shows scores similar to SDAC, but the cost rate converges above the threshold 0.025. In conclusion, SDAC can efficiently satisfy the safety constraints through the use of Q-functions and improve score performance through the distributional critics.

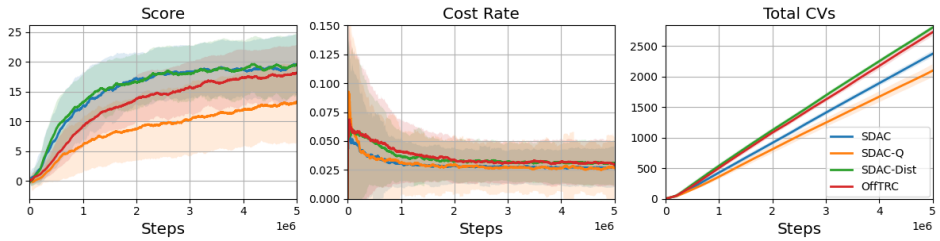
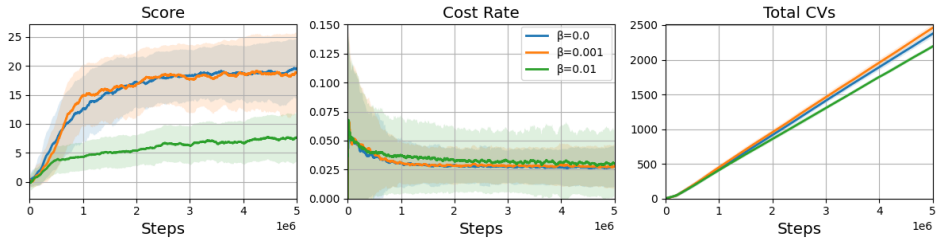


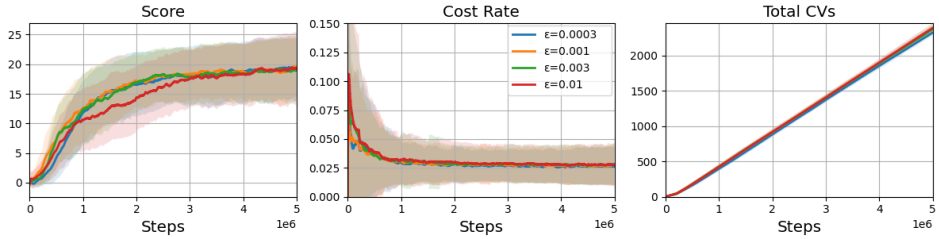
Figure 6: Training curves of variants of SDAC for the point goal task.

C.3 Ablation Study on Hyperparameters

To check the effects of the hyperparameters, we conduct ablation studies on the trust region size ϵ and entropy coefficient β. The results on the entropy coefficient are presented in Figure 7a, showing that the score significantly decreases when β is 0.01. This indicates that policies with high entropy fail to improve score performance since they focus on satisfying the constraints. Thus, the entropy coefficient should be adjusted cautiously, or it can be better to set the coefficient to zero. The results on the trust region size are shown in Figure 7b, which shows that the results do not change significantly regardless of the trust region size. However, the score convergence rate for $\epsilon = 0.01$ is the slowest because the estimation error of the surrogate increases as the trust region size increases according to Theorem A.8.



(a) Entropy coefficient β.



(b) Trust region size ϵ.

Figure 7: Training curves of SDAC with different hyperparameters for the point goal task.

D Comparison with RL Algorithms

In this section, we compare the proposed safe RL algorithm with traditional RL algorithms in the locomotion tasks and show that safe RL has the advantage of not requiring reward tuning. We use the truncated quantile critic (TQC) [Kuznetsov et al., 2020], a state-of-the-art algorithm in existing RL benchmarks [Todorov et al., 2012], as traditional RL baselines. To apply the same experiment to traditional RL, it is necessary to design a reward reflecting safety. We construct the reward through a weighted sum as $\bar{R} = (R - \sum_{i=1}^{3} w_i C_i)/(1 + \sum_{i=1}^{3} w_i)$, where R and $C_{\{1,2,3\}}$ are used to train safe RL methods and are defined in Appendix B, and $\bar{R}$ is called the *true reward*. The weights of the reward function $w_{\{1,2,3\}}$ are searched by a Bayesian optimization tool[1] to maximize the true reward of TQC in the Mini-Cheetah task. Among the 63 weights searched through Bayesian optimization, the top five weights are listed in Table 5.

Table 5: Weights of the reward function for the Mini-Cheetah task.

Reward weights	w_1	w_2	w_3
#1	1.588	0.299	0.174
#2	1.340	0.284	0.148
#3	1.841	0.545	0.951
#4	6.560	0.187	4.920
#5	1.603	0.448	0.564

Figure 8 shows the training curves of the Mini-Cheetah task experiments where TQC is trained using the weight pairs listed in Table 5. The graph shows that it is difficult for TQC to lower the second cost below the threshold while all costs of SDAC are below the threshold. In particular, TQC with the fifth weight pairs shows the lowest second cost rate, but the true reward sum is the lowest. This shows that it is challenging to obtain good task performance while satisfying the constraints through reward tuning.

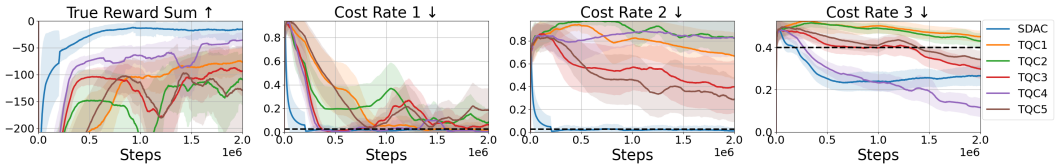


Figure 8: Training curves of the Mini-Cheetah task. The black dashed lines show the thresholds used for the safe RL method. The solid line represents the average value, and the shaded area shows one-fifth of the std value. The number after TQC in the legend indicates which of the reward weights in Table 5 is used. All methods are trained with five different random seeds.

[1]We use Sweeps from Weights & Biases Biewald [2020].

E Computational Cost Analysis

E.1 Complexity of Gradient Integration Method

In this section, we analyze the computational cost of the gradient integration method. The proposed gradient integration method has three subparts. First, it is required to calculate policy gradients of each cost surrogate, g_k, and $H^{-1}g_k$ for $\forall k \in \{1, 2, ..., K\}$, where H is the Hessian matrix of the KL divergence. $H^{-1}g_k$ can be computed using the conjugate gradient method, which requires only a constant number of back-propagation on the cost surrogate, so the computational cost can be expressed as $K \cdot O(\mathrm{BackProp})$.

Second, the quadratic problem in Section 3.1 is transformed to a dual problem, where the transformation process requires inner products between g_k and $H^{-1}g_m$ for $\forall k, m \in \{1, 2, ..., K\}$. The computational cost can be expressed as $K^2 \cdot O(\mathrm{InnerProd})$.

Finally, the transformed quadratic problem is solved in the dual space $\in \mathbb{R}^K$ using a quadratic programming solver. Since K is usually much smaller than the number of policy parameters, the computational cost almost negligible compared to the others. Then, the cost of the gradient integration is $K \cdot O(\mathrm{BackProp}) + K^2 \cdot O(\mathrm{InnerProd}) + C$. Since the back-propagation and the inner products is proportional to the number of policy parameters $|\psi|$, the computational cost can be simplified as $O(K^2 \cdot |\psi|)$.

E.2 Quantitative Analysis

Table 6: Training time of Safe RL algorithms (in hours). The training time of each algorithm is measured as the average time required for training with five random seeds. The total training steps are $5 \cdot 10^6$ and $3 \cdot 10^6$ for the point goal task and the Mini-Cheetah task, respectively.

Task	**SDAC (proposed)**	OffTRC	WCSAC	CPO	CVPO
Point goal (Safety Gym)	7.96	4.86	19.07	2.61	47.43
Mini-Cheetah (Locomotion)	8.36	6.54	16.41	1.99	-

We analyze the computational cost of the proposed method quantitatively. To do this, we measure the training time of the proposed method, SDAC, and the safe RL baselines. We use a workstation whose CPU is the Intel Xeon e5-2650 v3, and GPU is the NVIDIA GeForce GTX TITAN X. The results are presented in Table 6. While CPO is the fastest algorithm, its performance, such as the sum of rewards, is relatively poor compared to other algorithms. The main reason why CPO shows the fastest computation time is that CPO is an on-policy algorithm, hence, it does not require an insertion to (and deletion from) a replay memory, and batch sampling. SDAC shows the third fastest computation time in all algorithms and the second best one among off-policy algorithms. Especially, SDAC is slightly slower than OffTRC, which is the fastest one among off-policy algorithms. This result shows the benefit of SDAC since SDAC outperforms OffTRC in terms of the returns and CV, but the training time is not significantly increased over OffTRC. WCSAC, which is based on SAC, has a slower training time because it updates networks more frequently than other algorithms. CVPO, an EM-based safe RL algorithm, has the slowest training time. In the E-step of CVPO, a non-parametric policy is optimized to solve a local subproblem, and the optimization process requires discretizing the action space and solving a non-linear convex optimization for all batch states. Because of this, CVPO takes the longest to train an RL agent.

References

J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proceedings of International Conference on Machine Learning*, pages 22–31, 2017.

M. G. Bellemare, W. Dabney, and M. Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023. `http://www.distributional-rl.org`.

L. Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

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

J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1996.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of International Conference on Machine Learning*, pages 1861–1870, 2018.

D. Kim and S. Oh. Efficient off-policy safe reinforcement learning using trust region conditional value at risk. *IEEE Robotics and Automation Letters*, 7(3):7644–7651, 2022a.

D. Kim and S. Oh. TRC: Trust region conditional value at risk for safe reinforcement learning. *IEEE Robotics and Automation Letters*, 7(2):2621–2628, 2022b.

A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *Proceedings International Conference on Machine Learning*, pages 5556–5566, 2020.

J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020.

Z. Liu, Z. Cen, V. Isenbaev, W. Liu, S. Wu, B. Li, and D. Zhao. Constrained variational policy optimization for safe reinforcement learning. In *Proceedings of International Conference on Machine Learning*, pages 13644–13668, 2022.

W. Meng, Q. Zheng, Y. Shi, and G. Pan. An off-policy trust region policy optimization method with monotonic improvement guarantee for deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5):2223–2235, 2022.

T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

D. Precup, R. S. Sutton, and S. P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of International Conference on Machine Learning*, pages 759–766, 2000.

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

M. Rowland, M. Bellemare, W. Dabney, R. Munos, and Y. W. Teh. An analysis of categorical distributional reinforcement learning. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2018.

E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

Q. Yang, T. D. Simão, S. H. Tindemans, and M. T. J. Spaan. WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10639–10646, 2021.