2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# Risk-Aware Reinforcement Learning with Bandit-Based Adaptation for Quadrupedal Locomotion

Yuanhong Zeng* Anushri Dixit†

***Abstract*— In this work, we introduce a risk-aware reinforcement learning framework for robust quadrupedal locomotion. Our approach first trains a family of risk-conditioned policies using a Conditional Value-at-Risk (CVaR) constrained optimization technique, which improves both training stability and sample efficiency. During deployment, we frame online policy selection as a multi-armed bandit problem. Relying solely on observed episodic returns rather than privileged environment information, this method dynamically adjusts the robot's robustness level to handle unknown conditions on the fly. We evaluate our approach in simulation across eight diverse settings—varying dynamics, contacts, sensing noise, and terrain—as well as in real-world trials on a Unitree Go2 robot. Compared to existing baselines, our risk-aware policy achieves nearly twice the mean and tail performance in novel environments, with the bandit algorithm successfully identifying the optimal policy within just two minutes of operation.**

## I. Introduction

Quadrupedal robots excel in unstructured and hazardous terrains for tasks like mapping and search-and-rescue. While early controllers relied on model predictive control [1], which requires complex dynamics models, modern approaches leverage simulators like Isaac Sim [2] and massively parallelized model-free reinforcement learning [3] for rapid locomotion training.

A significant challenge in deploying policies trained in simulation onto real hardware is that the shift between training and deployment conditions may cause a difference in the performance expected versus what is attained in the real world. A common mitigation strategy is domain randomization [4], but selecting parameters and ranges that both stabilize training and produce good deployment performance requires substantial manual effort. Risk-aware RL methods offer another way to mitigate sim-to-real shift by emphasizing performance in the tail of the training distribution rather than maximizing expected returns [5]. However, these methods can be overly conservative and difficult to train, as they require specifying a desired risk (or robustness) level a priori. In practice, because the deployment environment is unknown, any chosen risk level is likely to be miscalibrated, leading to behaviors that are either too optimistic or too conservative.

We address these issues using CVaR-constrained optimization: maximizing the expected return subject to a CVaR constraint via a Lagrangian formulation. While prior work like CPPO [6] demonstrates that such constraints yield robust safety guarantees in perturbed environments, CPPO evaluates the constraint using raw Monte Carlo returns. In high-dimensional quadrupedal locomotion, this introduces prohibitive variance. We overcome this limitation by evaluating the CVaR constraint using a bootstrapped, truncated return formulation. This key distinction drastically reduces variance and seamlessly integrates with standard estimators like Generalized Advantage Estimation [7] and Proximal Policy Optimization (PPO) [8], enabling scalable and stable training.

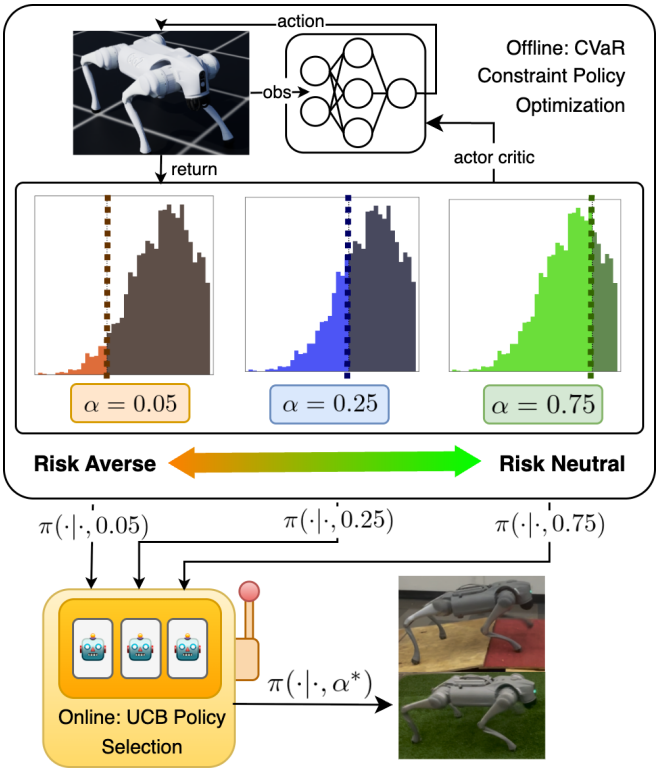


Fig. 1. Overview: We train multiple policies using CVaR-constrained policy optimization, where each critic focuses on a different tail of the return distribution, resulting in policies with varying levels of risk awareness. An upper confidence bound (UCB) bandit is then used to adaptively select the appropriate risk level online. The bandit selects the best performing policy over time as the robot interacts with the environment repeatedly.

As shown in Figure 1, we train a family of CVaR-constrained reinforcement learning policies at different risk-levels, $\alpha \in (0, 1)$. A risk-neutral policy may perform well in simple, flat ground but may fail on undulating terrain. On the other hand, a policy that is risk-averse may be successful

*Yuanhong Zeng is with the Department of Electrical and Computer Engineering, University of California Los Angeles, Los Angeles, CA, USA yuanhongzeng@ucla.edu
†Anushri Dixit is with the Department of Mechanical and Aerospace Engineering, University of California Los Angeles, Los Angeles, CA, USA anushridixit@ucla.edu

over rocky terrain but will be too slow or conservative on flat ground. Because deployment conditions can range from near-nominal to unstructured, we introduce an online policy selection mechanism using a multi-armed bandit that adaptively chooses the most appropriate risk-level for the current environment.

We evaluate our approach in both simulation and hardware. Our method achieves nearly twice the mean and tail performance of a PPO baseline trained in the same environment when deployed under disturbances, and the bandit selector converges to the best policy for an unknown test environment within two minutes of wall-clock time. Our contributions can be summarized as follows:

- We propose a framework for learning risk-aware policies via CVaR-constrained RL. Our RL algorithm augments PPO with a clipped-Lagrangian to account for the CVaR constraint. Compared to the original CPPO from [6], our method has improved training stability and sample efficiency.
- We present an online Upper Confidence Bound (UCB)-based multi-armed bandit that selects the risk level online to match and adapt to unknown environment conditions.
- We conduct comprehensive evaluation in simulation and on a Unitree Go2 Edu robot, demonstrating robust behavior under diverse disturbances and rapid convergence of the selection mechanism to the best policy.

## II. Related Works

### A. Distributional RL

Most risk-aware RL methods optimize a tail-risk objective directly. For example, [5] trains a quadruped by maximizing the CVaR of return, and [9] spans a spectrum of risk attitudes by optimizing various distortion risk measures, including CVaR and the Wang metric [10]. These approaches typically rely on distributional RL [11]–[13] to model the return distribution and design critics whose advantages reflect the chosen risk functional. More recent work CPO [14] performs CVaR optimization by capping high return trajectories during training. This approach can avoid explicit distribution modeling, yielding faster and more stable training. A drawback of directly optimizing CVaR is that the critic ignores or downweights high-return trajectories, which can reduce sample efficiency and increase the required environment interaction [15].

### B. Risk-Aware, Constrained RL

We build our work based on the constrained RL [16]–[18] framework. We highlight two benefits of the constrained approach. First, Ying et al. [6] show that the performance gap when deployed in an environment with different transition probability can be can be decreased by increasing the value at the worst state, and CVaR of return can be seen as a surrogate, which is bounded below when the constraint is satisfied. This results offers some performance guarantee on this method. Second, the gradient signal comes from both the expected return object and the CVaR constraint penalty. This yields better sample efficiency and training results compared to previously discussed methods only optimizes tailed return.

### C. Online Adaptation and Learning

Most state-of-the-art locomotion methods adaptively adjust policy behavior. A common approach is to condition the policy on environment features estimated from different modalities such as depth vision [19], proprioception [20], or LiDAR [21]. However, these methods usually assume access to privileged information during training (e.g., accurate height maps) and still depend on carefully designed training environments. An alternative line of research [5], [9], conditions policy behavior on the return distribution. Our work is closely related: we employ a UCB-based bandit to explore the return distribution and select actions that maximize its upper confidence bound. Bandit algorithms [22]–[24], provide a principled way to balance exploration and exploitation with theoretical regret guarantees [25], adapting directly from observed returns instead of relying on handcrafted features. Compared to heuristic or feature-matching methods, they require less privileged information.

## III. Preliminaries: Conditional Value-at-risk

For stochastic optimization problems, constraints can be reformulated by a commonly used risk measure called the *Value-at-Risk* (VaR). Given a confidence level $\alpha \in (0, 1)$, $\mathrm{VaR}_\alpha$ denotes the $\alpha$-quantile value of the return variable $R$ and is defined as,

$$\mathrm{VaR}_\alpha(R) := \inf\{\eta \,|\, \mathbb{P}(R \leq \eta) \geq \alpha\}.$$

It follows that $\mathrm{VaR}_\alpha(R) \geq 0 \implies \mathbb{P}(R \leq 0) \leq \alpha$. However, VaR is generally nonconvex and hard to compute. We now introduce a convex and monotonic risk measure called the conditional value-at-risk.

The *conditional value-at-risk*, $\mathrm{CVaR}_\alpha$, measures the expected loss in the $\alpha$-tail below the threshold $\mathrm{VaR}_\alpha$. $\mathrm{CVaR}_\alpha$ is computed as [26],

$$\mathrm{CVaR}_\alpha(R) := \sup_{\eta \in \mathbb{R}} \mathbb{E}\left[\eta - \frac{(\eta - R)^+}{\alpha}\right], \tag{1}$$

where $(\cdot)^+ := \max\{\cdot, 0\}$. A value of $\alpha \simeq 1$ corresponds to a risk-neutral case where $\mathrm{CVaR}_{\alpha \simeq 1}(R) \to \mathbb{E}[R]$ and a value of $\alpha \to 0^+$ is risk-averse. CVaR provides a convex lower bound of VaR, i.e.,

$$0 \leq \mathrm{CVaR}_\alpha(R) \leq \mathrm{VaR}_\alpha(R) \implies \mathbb{P}(R \leq 0) \leq \alpha. \tag{2}$$

## IV. Problem Statement

We model quadruped velocity-tracking as a discounted Markov Decision Process (MDP) $M := (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ with discount factor $\gamma \in (0, 1)$. The state space $\mathcal{S}$ comprises base linear and angular velocities, normalized projected gravity vector, velocity command, joint positions, joint velocities, and past action vector. $\mathcal{A}$ is the action space which consists of target joint positions. $P$ is the state transition kernel and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. We assume the

rewards are bounded. For a stationary policy $\pi$ under MDP, $M$, the $\alpha$-tailed return, $\alpha \in (0, 1]$, is defined as,

$$J(\pi; \alpha, M) := \text{CVaR}_\alpha \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]$$

where $a_t \sim \pi(\cdot \mid s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$.

We define $J(\pi) := J(\pi; 1, M)$ to be the expected value of the sum of the discounted reward of policy $\pi$ when deployed in $M$, and $D(\xi) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$ to be the sum of discounted reward of the trajectory $\xi = (s_0, a_1, s_1, \ldots)$.

For a policy is deployed in an unknown environment, we define a perturbed MDP $\widehat{M} = (\mathcal{S}, \mathcal{A}, \widehat{P}, R, \gamma)$ with the same state space, action space, and reward but a shifted transition kernel $\widehat{P}$.

**Problem 1** For the MDP $\widehat{M}$, we seek a risk-conditioned policy $\pi_\theta(a|s, \alpha)$, with risk parameter $\alpha \in (0, 1]$ and policy parameter $\theta$ by maximizes the $\alpha$-tailed return under an unknown transition kernel $\widehat{P}$.

$$\theta^* = \arg\max_\theta J(\pi_\theta; \alpha, \widehat{M}) \tag{3}$$

**Problem 2** Suppose we have a finite set of policies $\{\pi_{\theta_k}(\cdot|\cdot, \alpha_k)\}_{k=1}^K$, each trained with a distinct risk level $\alpha_k$. We formulate online policy selection problem as a stochastic multi-arm bandit over episode problem [25]. At episode $e$, the learner selects an arm in $k \in \{1, \ldots, K\}$, executes $\pi_{\theta_k}$ for $T$ steps in $\widehat{M}$, and then observes the undiscounted episodic return $X_{k,e} = \sum_{t=0}^{T-1} R(s_t, a_t)$. For each arm $k$, we assume that $\{X_{k,e}\}_{e=1}^H$ are $H$ i.i.d draws from a stationary distribution with bounded mean and variance. The goal is to minimize regret defined as

$$\mathcal{R}(H; \alpha) := H \cdot \mathbb{E}X_{k^*} - \sum_{e=1}^{H} X_{k_e, e} \tag{4}$$

where, $k_e$ is the arm selected at episode $e$, and $k^* = \arg\max_k \mathbb{E}X_k$.

## V. Method

### A. CVaR Constrained Proximal Policy Optimization

Unlike recent methods based on return capping [14] and distributional RL [5], [11], [13], [27] that attempt to solve Problem 1 by directly maximizing the CVaR of return, we propose to solve the CVaR constrained problem,

$$\begin{aligned} \max_\theta \quad & \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \hat{P}(\cdot|s_t, a_t)} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right] \\ \text{s.t} \quad & J(\pi; \alpha, \widehat{M}) \geq \beta. \end{aligned} \tag{5}$$

Since samples from $\widehat{P}$ are not available, we use samples from $P$ to approximate the corresponding expectations. We first convert the constrained problem in (5) into an unconstrained form via a Lagrangian relaxation [28] and using the CVaR definition in (1),

$$\begin{aligned} \max_{\lambda \geq 0} \min_{\theta, \eta} L(\theta, \eta, \lambda) := & -J(\pi_\theta) \\ & + \lambda \left( \beta - \eta + \frac{1}{\alpha} \mathbb{E}_{\xi \sim \pi_\theta} \left[ (\eta - D(\xi))^+ \right] \right). \end{aligned} \tag{6}$$

To solve (6), we use the following gradients from [29],

$$\nabla_\eta L = -\lambda + \frac{\lambda}{\alpha} \mathbb{E}_{\xi \sim \pi_\theta} \mathbf{1}\{D(\xi) \leq \eta\}, \tag{7}$$

$$\nabla_\theta L = -\mathbb{E}_{\xi \sim \pi_\theta} \left[ \nabla_\theta \log P_\theta(\xi) \right] \left( D(\xi) - \frac{\lambda}{\alpha} (\eta - D(\xi))^+ \right), \tag{8}$$

$$\nabla_\lambda L = \beta - \eta + \frac{1}{\alpha} \mathbb{E}_{\xi \sim \pi_\theta} \left[ (\eta - D(\xi))^+ \right]. \tag{9}$$

Although Chow et al. show that gradient descent using (7)-(9) converges to locally optimal policies in [29], in practice directly updating the policy with (8) is fragile. We instead update $\theta$ by minimizing,

$$\begin{aligned} & L_{\text{PPO-Lagrangian}} \\ & := -\mathbb{E}_{(s,a) \sim \pi_{\theta_{\text{old}}}} \left[ \min \left( r A(s, a), g(\epsilon, A(s, a)) \right) \right] \\ & + \frac{\lambda}{\alpha} \mathbb{E}_{\xi_{\text{T}} \sim \pi_{\theta_{\text{old}}}} \left[ \min \left( r(\eta - \widetilde{D}(\xi_{\text{T}}))^+, g(\delta, (\eta - \widetilde{D}(\xi_{\text{T}}))^+) \right) \right] \end{aligned} \tag{10}$$

$$r = \frac{\pi_\theta(s|a; \alpha)}{\pi_{\theta_{\text{old}}}(s|a; \alpha)} \tag{11}$$

$$g(\epsilon, A) = \begin{cases} (1 + \epsilon)A, & \text{if } A \geq 0 \\ (1 - \epsilon)A, & \text{if } A < 0, \end{cases} \tag{12}$$

$$\widetilde{D}(\xi_T) = \sum_{(a_t, s_t) \in \xi_T} \gamma^t R(s_t, a_t) + \gamma^T V_{M, \pi_{\theta_{\text{old}}}}(s_T), \tag{13}$$

where, $\theta_{\text{old}}$ is the parameter of the policy used to collect current on-policy batch, $A$ is the generalized advantage estimates described in Eq 16 [7]. $\epsilon$ and $\delta$ are the clipping thresholds for the PPO loss and the Lagrangian respectively. Here, (10) augments the PPO clipped surrogate with the clipped Lagrangian term; We use the gradient of (10) to update $\theta$ and use the gradient calculated in (7) and (9) to update $\eta$ and $\lambda$ to reflect the true constraint geometry. The complete procedure is described in Algorithm 1.

The design of our algorithm specifically addresses the limitations of applying prior constrained-CVaR methods, such as CPPO [6], to high-dimensional quadrupedal locomotion. While [6] relies on raw Monte Carlo (MC) returns to evaluate the CVaR constraint, this approach suffers from large variance and requires full episodic rollouts, precluding the use of modern massively parallel simulators [3]. To overcome these bottlenecks, our approach is motivated by three considerations. First, we replace full MC rollouts with a bootstrapped return $\widetilde{D}(\xi_T)$ estimated from truncated length-$T$ trajectories. This drastically reduces variance and enables efficient, highly parallelized rollout generation. Second, to further stabilize gradient updates, we adopt PPO with generalized advantage estimation [7]. Third, large unconstrained

steps can cause catastrophic performance drops [16]; PPO-style clipping prevents extreme tail returns from producing unbounded penalty gradients when the risk threshold $\alpha$ is small.

**Algorithm 1** Offline: CVaR Policy Optimization

**Require:** hyperparameters $\alpha$, $\epsilon$, $\delta$, $\lambda_{\max}$ learning rates $lr_\eta, lr_\theta, lr_\lambda$, function $f$ for updating $\beta$, rollout length $T$

**Ensure:** policy $\pi_\theta$

**for** $j = 1, 2, \ldots, N_{\text{iter}}$ **do**
  Generate $N$ trajectories $\{\xi_T^{(i)}\}_{i=1}^N$ with $\pi_{\theta_{\text{old}}}$.
  Compute $A(s_t, a_t)$ and $\widetilde{D}(\xi_T^{(i)})$ $\forall i,\ \forall (s_t, a_t) \in \xi_T^{(i)}$
  Calculate $\nabla_\theta L_{\text{PPO-Lagrangian}}$
  $\eta \leftarrow \eta - lr_\eta \cdot \nabla_\eta L$
  $\theta \leftarrow \theta - lr_\theta \cdot \nabla_\theta L_{\text{PPO-Lagrangian}}$
  $\lambda \leftarrow \lambda + lr_\lambda \cdot \nabla_\lambda L$
  $\lambda \leftarrow \min(\max(0, \lambda), \lambda_{\max})$
  $\beta \leftarrow f(\{\xi_T^{(i)}\}_{i=1}^N)$
  $\theta_{\text{old}} \leftarrow \theta$
**end for**

### B. Bandit-based Online Policy Selection

Choosing the risk level $\alpha$ trades off safety and performance: very small $\alpha$ can be overly conservative and vice versa. We select $\alpha$ by choosing the policy that maximizes episodic (undiscounted) return in the target environment. Because we lack privileged information (e.g., height maps, friction), we assume only access to the realized reward during interaction, computable online from observations and proprioception, so policy selection reduces to a stochastic bandit problem (Sec. IV).

We use the Empirical-Bernstein UCB [23], [30], which is tighter than Hoeffding-based UCB [25] and typically converges faster. At episode $e$, for each arm $k$ we compute.

$$\text{UCB}_k(e) := \hat{\mu}_k(e) + \sqrt{\frac{2\,\hat{\sigma}_k(e)\,\ln\!\big(3/\delta_e\big)}{N_k(e)}}\,\frac{3R\,\ln\!\big(3/\delta_e\big)}{N_k(e)}, \tag{14}$$

and select the arm with the largest UCB value. Here, $N_k(e)$ is the number of pulls of arm $k$, $\hat{\mu}_k(e) = \frac{1}{N_k(e)} \sum_{l=1}^{e} X_{k,l}$ and $\hat{\sigma}_k(e) = \frac{1}{N_k(e)-1} \sum_{l=1}^{e} (X_{k,l} - \hat{\mu}_k(e))^2$ are the sample mean and variance of the episodic return of arm $k$, $R$ is the reward range, and $\delta_e = \delta/(Ke^2)$. The description of the selection algorithm is provided in Algorithm 2.

**Algorithm 2** Online: Empirical–Bernstein UCB

**Require:** policies with different $\alpha$-level$\{\pi_{\theta_k}(\cdot|\cdot, \alpha_k)\}_{k=1}^K$, reward range $R$, episode length $T$

Initialize selection counter $N_k \leftarrow 0$
**for** $k = 1$ to $K$ **do**
  Run $\pi_{\theta_k}$ for $T$ step; save episodic return $x$
  Calculate $\hat{\mu}_k$ and $\hat{\sigma}_k$ from history of $x$
**end for**
**for** $e = K + 1$ to $E$ **do**
  $\delta_e \leftarrow \delta/(K\,e^2)$
  **for** $k = 1$ to $K$ **do**
    Calculate $\hat{\sigma}_k$ and $\hat{\mu}_k$ from history of $x$
    $U_k \leftarrow \text{UCB}\big(\hat{\mu}_k,\, \hat{\sigma}_k,\, N_k,\, R,\, \delta_e\big)$
  **end for**
  $k^\star \leftarrow \arg\max_k U_k$
  $N_{k^*} \leftarrow N_{k^*} + 1$
  Run policy $\pi_{\theta_k}$ for $T$ step; save episodic return $x$
**end for**

## VI. Simulation Experiments

In this section, we validate our RL framework in simulation: Section VI-A studies the training behavior when compared to other robust RL approaches based on CVaR optimization. Section VI-B investigates the tradeoff of different $\alpha$ value when deployed to perturbed environments.

**Training Setup** We built our policy based on skrl [31]. The actor and critic network are implemented as multi-layer perceptrons with hidden layers of dimension $[256, 256, 256]$ and Exponential Linear Unit (ELU) activation. As the return distribution of the policy improves during training, we adaptively change the CVaR constraint level, $\beta$. We set $\beta$ to the exponential moving average of the current $\text{VaR}_\alpha$ using,

$$\beta_t \leftarrow 0.3\text{VaR}_{\xi\sim\pi}^{\alpha}(\widetilde{D}(\xi_T)) + 0.7\beta_{t-1}. \tag{15}$$

Given that, $\text{VaR}_\alpha \geq \text{CVaR}_\alpha$, this schedule keeps the constraint active, progressively shifting probability mass out of the lower tail. We trained the policy in Isaac Lab [2] in a flat environment with reward terms defined in Table I and domain randomization added to distort the observation and randomize joint gain. We describe the domain randomization parameters in Table II. We also set up a curriculum with gradually increasing command magnitude for stable training.

TABLE I
Reward terms used in training.

| Reward Term | Definition | Weight |
|---|---|---|
| track linear velocity xy | $\exp(-\lVert\mathbf{v}_{xy}^{\text{cmd}} - \mathbf{v}_{xy}\rVert^2/\sigma)$ | 1.5 |
| track angular velocity yaw | $\exp(-(\omega_{\text{yaw}}^{\text{cmd}} - \omega_{\text{yaw}})^2/\sigma)$ | 0.75 |
| linear velocity z l2 | $v_z^2$ | $-1.0$ |
| angular velocity xy l2 | $\lVert\omega_{\text{roll,pitch}}\rVert^2$ | $-0.05$ |
| joint acceleration l2 | $\lVert\ddot{\mathbf{q}}\rVert^2$ | $-2.5 \times 10^{-7}$ |
| torques l2 | $\lVert\boldsymbol{\tau}\rVert^2$ | $-2 \times 10^{-4}$ |
| action rate l2 | $\lVert\mathbf{a}\rVert^2$ | $-0.05$ |
| feet air time | $\sum_{f=1}^{4}(t_{\text{air},f} - 0.5)$ | 0.35 |
| flat orientation l2 | $\lVert\phi_{\text{roll,pitch}}\rVert^2$ | $-2.0$ |

We trained 2048 agents in parallel for 30,000 steps within the environment. This approximately takes 15 minutes wall clock time on a workstation with 24 Core AMD-5900X CPU with a NVIDIA A5000 GPU.

### A. Algorithm Performance

In these experiments, we demonstrate the training performance of our method, Algorithm 1, by comparing with multiple risk-aware RL algorithms. We compare our method to **PPO** [8]; **CPPO** [6], a simplified variant of ours

TABLE II

DOMAIN RANDOMIZATION PARAMETERS

| Randomization Parameter | Range |
|---|---|
| joint gain | $[22, 27]$ |
| joint damping | $[0.3, 0.7]$ |
| noise added to base velocity obs | $[-0.1, 0.1]$ |
| noise added to ang velocity obs | $[-0.2, 0.2]$ |
| noise added to projected gravity | $[-0.05, 0.05]$ |
| noise added to joint velocity obs | $[-1.5, 1.5]$ |


Fig. 2. Training performance at $\alpha = 0.25$. Curves show mean undiscounted return versus timesteps of environment interactions; solid lines are the mean over 10 runs trained for 30,000 timesteps, shaded regions denote $\pm 1$ standard deviation. Evaluations are in the training environment.

without clipping or bootstrapped–return estimation; **return-capped PPO** [14], which directly optimizes Problem 3; and **DPPO** [9], which optimizes the same objective by shaping the value-distribution with QR-DQN [12] and SR($\lambda$) [32]. All baselines use the same training setup, hyperparameters, and architecture; for DPPO the critic head is replaced with a quantile head. We use CVaR at level $\alpha = 0.25$.

Figure 2 shows that our method learns more slowly at the start, due to optimizing the Lagrangian term, but attains comparable performance as PPO at the end. CPPO is unstable and slower to converge, motivating the clipping and bootstrapping components in our design. Capped PPO and DPPO, which both directly target the CVaR objective, display similar behavior for much of training; because they de-emphasize high-return trajectories, they tend to learn less from successes.

### B. Robustness Under Disturbances

In the previous subsection, we compared the performance of our method, Algorithm 1, to various baselines within the training distribution. Next, we study the robustness–performance trade-off across risk levels to show the need for online policy selection in unknown environments. We train six policies: five CVaR policies with $\alpha \in \{0.05, 0.1, 0.25, 0.5, 0.75\}$ and a PPO baseline. Each policy is evaluated in eight out-of-distribution environments that emulate shifts in low-level control, sensing/estimation, terrain, and collisions (Table III).

TABLE III

ENVIRONMENT DESCRIPTION

| Environment | Description |
|---|---|
| Brownian | Brownian noise with drift $10^{-3}$ added to linear velocity observations |
| Delay | Joint command delayed by 0.05s |
| Rough | Rough terrain with maximum height difference 10cm |
| Push | Random xy linear velocity added to base every 5s |
| Friction | Randomize static friction in range [0.3,0.7] |
| Jitter | Replace current observations with past observations with probability 0.6 |
| Gain | Randomize gain in range [10, 50] |
| Incline | The ground plane has a 20 degree incline |

As seen in Figure 3, across the first four settings (Brownian, Delay, Rough, Push), smaller $\alpha$ yields higher mean and tail performance; in Brownian and Delay, the $\alpha = 0.05$ policy achieves nearly twice the mean and CVaR of PPO, consistent with robustness guarantees for small $\alpha$. The trade-off is evident in the remaining settings: $\alpha = 0.25$ performs best on Friction and Jitter, while PPO leads on Gain and Incline, two environment conditions closest to training. Thus, neither the most conservative policy nor nominal PPO is uniformly optimal. This motivates our design of an online and adaptive policy selection scheme, Algorithm 2.

## VII. HARDWARE EXPERIMENTS

We now validate our offline training and online policy selection on a Unitree Go2 quadruped. Section VII-A compares the real-world performance at different values of $\alpha$. Section VII-B validates the performance of our policy selection algorithm based on UCB bandits.

**Hardware Setup** We deployed our policy on a Unitree Go2 Edu robot. The policy sends target joint position commands at 50Hz and the commands are tracked by the robot's PD controller with proportional gain, $k_p = 25$, and derivative gain, $k_d = 0.5$. We use the OptiTrack motion capture system to estimate the state of the robot. Because the motion capture system provides unrealistically accurate state estimation, we intentionally inject noise, drift, and latency during hardware evaluation. This simulates the degraded performance of typical onboard visual-inertial odometry (VIO) systems to rigorously test the policy's real-world robustness.

### A. Sim-to-real Robustness

We deployed the our policies trained at $\alpha = 0.05$ and $\alpha = 0.25$, and the PPO policy on a Unitree Go2 to assess real-world performance. Three settings were used. **Flat:** the robot was commanded to walk back and forth on flat ground. We did not introduce any disturbances in this environment. **Ramp:** the robot was commanded to traverse an unseen ramp at constant speed $v = 0.7\mathrm{m/s}$, introducing observation shifts, external forces, and unexpected contacts. **Grass:** the robot walked on turf with 11 cm soft soccer cones hidden beneath to simulate rough grassland; a trial was successful if the robot remained on its feet while tracking commands for 80s.

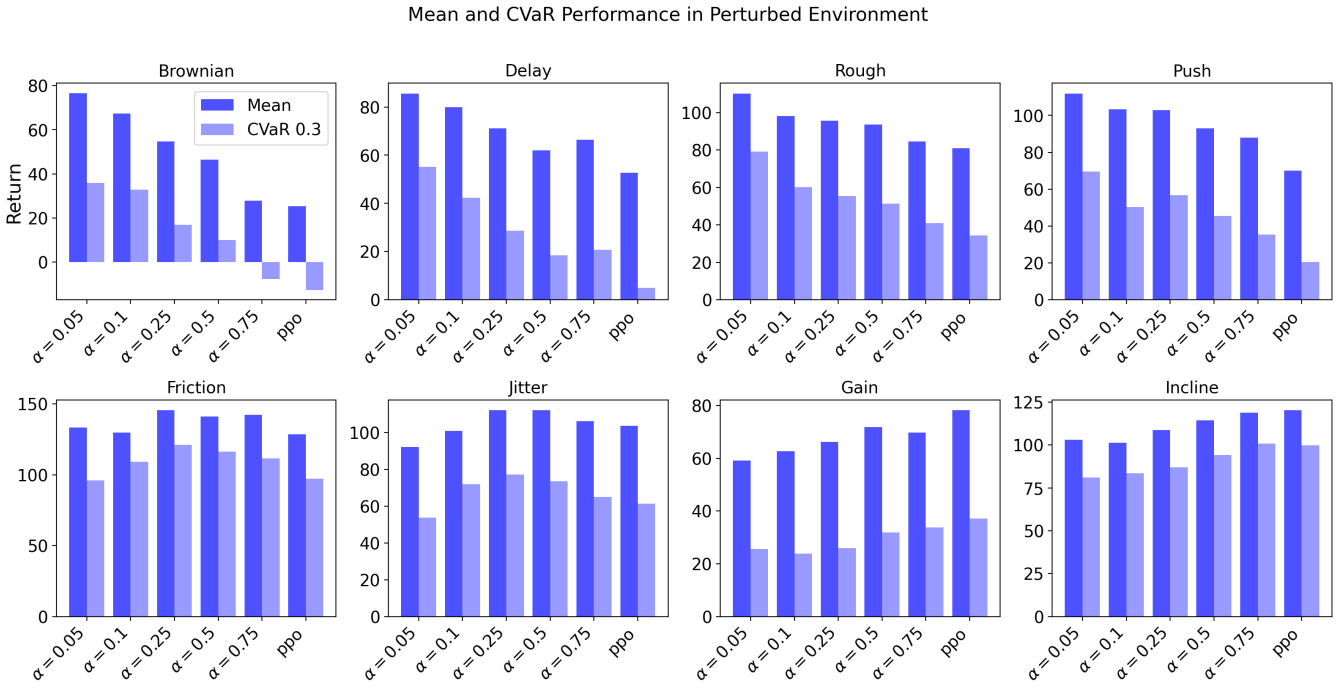


Fig. 3. Performance of with different $\alpha$-level. We trained policies different $\alpha$-level and test their performances in perturbed environments. The descriptions of the environments are available at Table III. For each $\alpha$-level we trained 10 policies using different seeds, and we evaluate each policy with 1000 parallel simulations and report the mean return over 6000 timesteps. We reset the environment and randomize the state of robots every 100 timesteps

We measured success rate, time-to-live (TTL), and reward rate. We conducted 15 experiments for each $\alpha$-level in each environment. Results are in Table IV; qualitative behaviors at $v = 1\,\mathrm{m/s}$ are illustrated in Fig. 4.

On flat ground, we observed that the PPO moved at a faster and uniform gait compared to $\alpha = 0.05$ and $\alpha = 0.25$, giving larger return. On the ramp, the $\alpha = 0.05$ policy advanced with shorter steps and was unable to climb 30% of the times (Fig. 4), which explains the longer completion times. The $\alpha = 0.25$ policy developed a stable trotting gait and achieved the best overall performance (highest success and per step reward), with most failures due to occasional drift off the ramp when not precisely following commands. PPO can be unstable and loss balance when unexpected contact happens, resulting in the lowest success rate. On grass, we observed a similar trade-off: PPO achieved the highest reward rate but was fragile to disturbances; $\alpha = 0.05$ attained the best success rate and longest TTL by maintaining stability in this unseen condition.

TABLE IV

POLICY PERFORMANCE ACROSS THREE ENVIRONMENTS.

| Experiment | Policy | Time/TTL (s) | Succ (%) | Reward/sec |
|---|---|---|---|---|
| Flat | $\alpha = 0.05$ | $18.0 \pm 5.0$ | 100 | $1.05 \pm 0.20$ |
| | $\alpha = 0.25$ | $15.0 \pm 7.0$ | 100 | $1.31 \pm 0.25$ |
| | PPO | $\mathbf{12.0 \pm 6.0}$ | 100 | $\mathbf{1.35 \pm 0.30}$ |
| Ramp | $\alpha = 0.05$ | $19.0 \pm 8.0$ | 55 | $0.95 \pm 0.35$ |
| | $\alpha = 0.25$ | $12.5 \pm 6.4$ | **70** | $\mathbf{1.32 \pm 0.32}$ |
| | PPO | $\mathbf{7.4 \pm 6.3}$ | 30 | $1.18 \pm 0.46$ |
| Grass | $\alpha = 0.05$ | $\mathbf{77.0 \pm 9.4}$ | **85** | $1.01 \pm 0.13$ |
| | $\alpha = 0.25$ | $59.2 \pm 28.1$ | 60 | $\mathbf{1.14 \pm 0.32}$ |
| | PPO | $64.2 \pm 36.0$ | 55 | $1.09 \pm 0.44$ |

### B. Online Policy Selection and Regret

In this experiment, we evaluate the convergence of the multi-arm bandit when deploying the robot across three testing environments. Using Algorithm 2 with an episode length of $T = 10$, the bandit selects among $\alpha = 0.05$, $\alpha = 0.25$, and PPO as the robot walks on flat ground, traverses a ramp, and moves over grass. We conducted seven trials per environment and recorded both the collected rewards and the selection history.

Table V shows how the selection probabilities evolve over time. In the Flat and Ramp experiments, the probability of choosing the highest-reward policy steadily increases and converges to 1 within 5000 timesteps (about 2 minutes). Figure 5 illustrates that this convergence often happens quickly after the warm-up rounds in the first 500 timesteps. By contrast, in the Grass experiment the convergence is less clear because the two higher-reward policies have similar means and high variance. Here, the bandit reliably rules out $\alpha = 0.05$ but splits its choices between the remaining two.

## VIII. CONCLUSION AND LIMITATIONS

We present a method for generating policies with different risk profiles using CVaR-constrained reinforcement learning. To adapt risk levels online, we introduce a simple UCB-bandit selection mechanism that adjusts based on the observed rewards during task execution. Our risk-aware

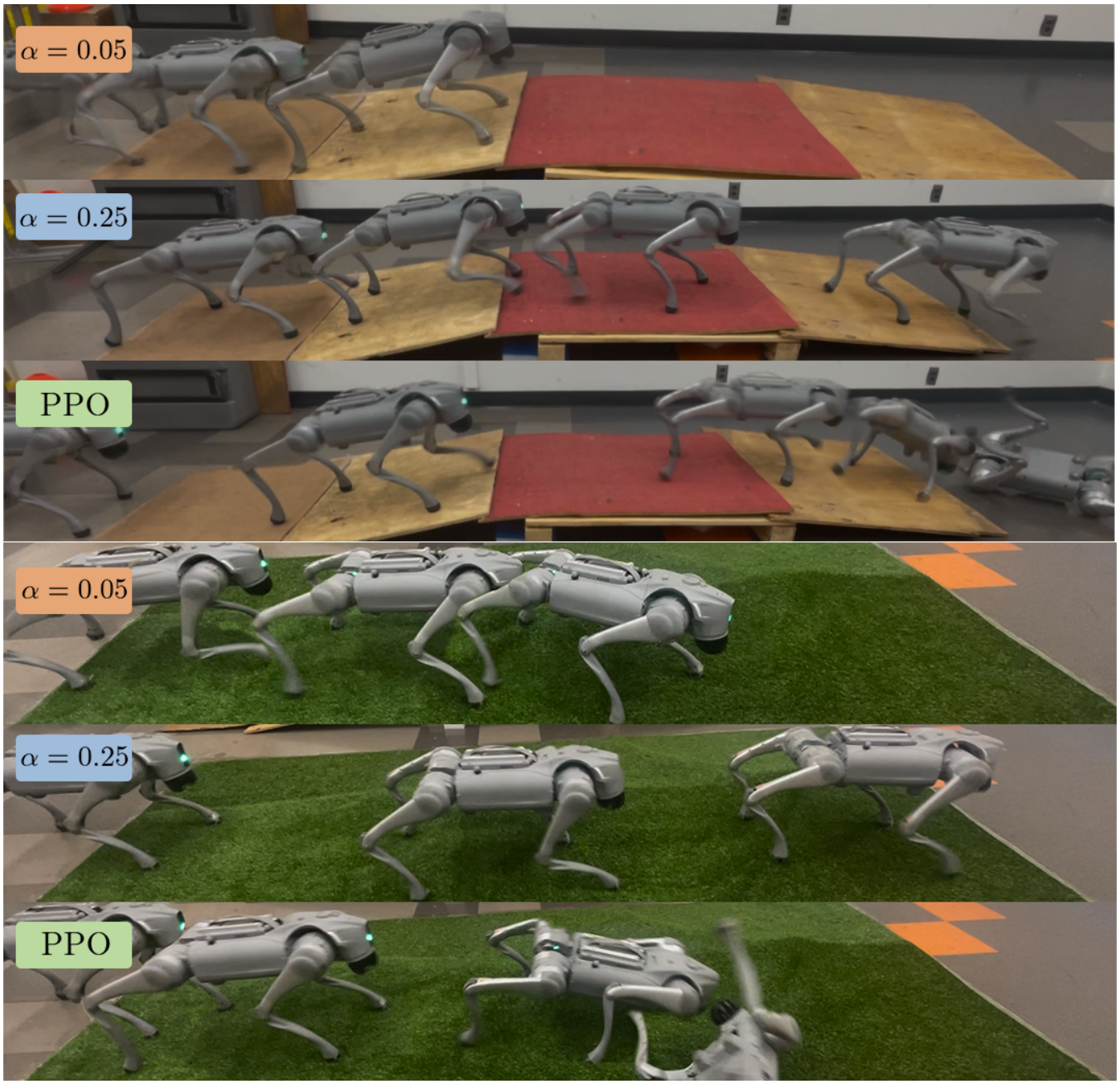


Fig. 4. We commanded the robot to walk at full speed ($v = 1$ m/s) on both ramp and grass terrains. The $\alpha = 0.05$ policy did not advance despite forward commands, either when climbing the ramp or when a foot became trapped in a soccer cone hidden beneath the grass. In contrast, the $\alpha = 0.25$ policy adapted a stable gait with larger steps, enabling the robot to step over hidden cones. The PPO baseline moved faster but was more prone to losing balance, particularly when descending the ramp or when one of its foot hits the soccer cone.

policies show robustness across environments with diverse disturbances, and the bandit algorithm converges quickly to the best-performing policy.

We identify three limitations for future work: (1) Our assumption of stable returns may fail in nonstationary settings, which windowed UCB could mitigate at the cost of extra tuning. (2) Relying solely on training rewards ignores execution risk; future deployments should incorporate explicit risk measures to favor safer policies. (3) The bandit is inherently ex-post, taking risky actions before updating beliefs. Integrating ex-ante risk predictors via reachability analysis [33] could resolve this.

## IX. Acknowledgements

We thank our UCLA colleagues—Allen Emmanuel Binny, Debajyoti Chakrabarti, and Qizhao Chen—for helpful feedback and support. We also thank David Martinez for OptiTrack system assistance and Professor Ankur Mehta for insightful guidance on improving the project.

TABLE V

SELECTION PROBABILITY BY TIMESTEP FOR THREE POLICIES.

| Experiment | Policy | 1250 | 2500 | 3750 | 5000 |
|---|---|---|---|---|---|
| Flat | $\alpha = 0.05$ | 0.10 | 0.11 | 0.10 | 0.00 |
| | $\alpha = 0.25$ | 0.20 | 0.19 | 0.10 | 0.00 |
| | PPO | 0.70 | 0.70 | 0.80 | 1.00 |
| Ramp | $\alpha = 0.05$ | 0.07 | 0.05 | 0.05 | 0.00 |
| | $\alpha = 0.25$ | 0.38 | 0.75 | 0.77 | 1.00 |
| | PPO | 0.54 | 0.20 | 0.17 | 0.00 |
| Grass | $\alpha = 0.05$ | 0.15 | 0.10 | 0.00 | 0.00 |
| | $\alpha = 0.25$ | 0.50 | 0.53 | 0.75 | 0.53 |
| | PPO | 0.34 | 0.37 | 0.25 | 0.47 |

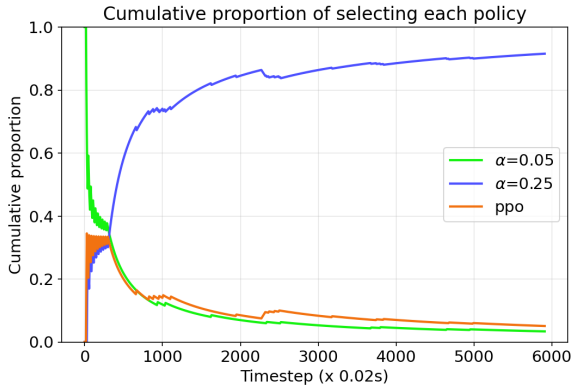


Fig. 5. Cumulative probability of selecting each policy in one ramp experiment. Selection of policy with $\alpha = 0.25$ quickly dominates after 2000 timesteps, which is roughly 1 minute wall clock time.

## REFERENCES


[1] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[2] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[3] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on robot learning*. PMLR, 2022, pp. 91–100.

[4] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[5] J. Shi, C. Bai, H. He, L. Han, D. Wang, B. Zhao, M. Zhao, X. Li, and X. Li, "Robust quadrupedal locomotion via risk-averse policy learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 459–11 466.

[6] C. Ying, X. Zhou, H. Su, D. Yan, N. Chen, and J. Zhu, "Towards safe reinforcement learning via constraining conditional value-at-risk," *International Joint Conference on Artificial Intelligence*, pp. 3673–3680, 2022.

[7] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

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

[9] L. Schneider, J. Frey, T. Miki, and M. Hutter, "Learning risk-aware quadrupedal locomotion using distributional reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 451–11 458.

[10] S. S. Wang, "A class of distortion operators for pricing financial and insurance risks," *Journal of risk and insurance*, pp. 15–36, 2000.

[11] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International conference on machine learning*. PMLR, 2017, pp. 449–458.

[12] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[13] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 1096–1105.

[14] H. Mead, C. Costen, B. Lacerda, and N. Hawes, "Return capping: Sample-efficient cvar policy gradient optimisation," *arXiv preprint arXiv:2504.20887*, 2025.

[15] I. Greenberg, Y. Chow, M. Ghavamzadeh, and S. Mannor, "Efficient risk-averse reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 639–32 652, 2022.

[16] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[17] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[18] S. Miryoosefi, K. Brantley, H. Daume III, M. Dudik, and R. E. Schapire, "Reinforcement learning with convex constraints," *Advances in neural information processing systems*, vol. 32, 2019.

[19] R. Yang, G. Yang, and X. Wang, "Neural volumetric memory for visual locomotion control," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 1430–1440.

[20] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[21] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," *arXiv preprint arXiv:2107.03996*, 2021.

[22] A. Tamkin, R. Keramati, C. Dann, and E. Brunskill, "Distributionally-aware exploration for cvar bandits," in *NeurIPS 2019 Workshop on Safety and Robustness on Decision Making*, 2019.

[23] V. Mnih, C. Szepesvári, and J.-Y. Audibert, "Empirical bernstein stopping," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 672–679.

[24] D. Bouneffouf, I. Rish, and C. Aggarwal, "Survey on applications of multi-armed and contextual bandits," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[25] A. Slivkins, *Introduction to Multi-Armed Bandits*. Foundations and Trends in Machine Learning, 2019.

[26] R. T. Rockafellar, S. Uryasev *et al.*, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000.

[27] X. Ma, J. Chen, L. Xia, J. Yang, Q. Zhao, and Z. Zhou, "Dsac: Distributional soft actor-critic for risk-sensitive reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 83, 2025.

[28] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[29] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *Journal of Machine Learning Research*, vol. 18, no. 167, pp. 1–51, 2018. [Online]. Available: http://jmlr.org/papers/v18/15-636.html

[30] A. Maurer and M. Pontil, "Empirical bernstein bounds and sample variance penalization," *arXiv preprint arXiv:0907.3740*, 2009.

[31] A. Serrano-Muñoz, D. Chrysostomou, S. Bøgh, and N. Arana-Arexolaleiba, "skrl: Modular and flexible library for reinforcement learning," *Journal of Machine Learning Research*, vol. 24, no. 254, pp. 1–9, 2023. [Online]. Available: http://jmlr.org/papers/v24/23-0112.html

[32] D. W. Nam, Y. Kim, and C. Y. Park, "Gmac: A distributional perspective on actor-critic framework," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7927–7936.

[33] I. Jang, J. Park, C. E. Mballo, S. Cho, C. J. Tomlin, and H. J. Kim, "Eigensafe: A spectral framework for learning-based stochastic safety filtering," *arXiv preprint arXiv:2509.17750*, 2025.