JOINT-RISK AWARE CONSTRAINED RISK-SENSITIVE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Maximizing rewards under various cost constraints is a frequent requirement when applying reinforcement learning (RL) to domains such as robotics and energy systems. Since expected-cost constraints alone cannot prevent rare but severe violations, constrained risk-sensitive RL is introduced to address this limitation through constraints on the risk of cost returns. However, because these constraints assess each cost independently using marginal distributions, they enforce overly conservative trade-offs by penalizing physically impossible events. We propose JRCPO, which preserves expected-cost constraints while replacing disjoint marginal risk constraints into a joint risk constraint over the reward-cost distribution. Utilizing the coherent duality of the Conditional Value at Risk (CVaR), JRCPO evaluates the CVaR of a scalarized objective along the Lagrange multiplier-directed trade-off path. This approach captures the joint distribution and considers only physically realizable events, resulting in an efficient reward-cost trade-off. Consequently, we demonstrate via evaluations on multi-cost tasks that the proposed method achieves higher reward returns than methods with cost-wise independent marginal risk constraints, and lower cost returns than expectation-based methods.

1 INTRODUCTION

Reinforcement learning (RL) frequently demands multivariate constrained settings when deployed in applied domains such as robotics (Liu et al., 2022; Thananjeyan et al., 2021; Chane-Sane et al., 2024), self-driving (Wang et al., 2025), or energy systems (Wang et al., 2023). For instance, a spider robot on a board may be required to be constrained by energy consumption and contact force with smooth locomotion. In such scenarios, the RL agent must optimize its primary objective (e.g., locomotion) while simultaneously satisfying multiple, often conflicting operational constraints (e.g., energy consumption and contact force). To solve the problem with RL, the constrained Markov Decision Process (CMDP) is often introduced(Altman, 1999).

Furthermore, to prevent the cost from rarely but severely exceeding its limit, the literature has integrated risk-sensitive RL and constrained RL (Yang et al., 2023; Enkhsaikhan & Jo, 2025; Kim et al., 2023; 2024). In this paper, risk refers to a real number that captures the severity of adverse outcomes stemming from the randomness of costs. By mitigating the risk of the constrained cost, one can expect the RL agent to more reliably avoid constraint violations. Specifically, Kim et al. (2024) successfully achieved very low constraint violations by constraining risks such as Conditional Value at Risk (CVaR). These approaches, however, associate the risk with each marginal cost, meaning that the RL agent may implicitly learn under the assumption of physically impossible events.

In the aforementioned spider robot example, a state with both high energy consumption and high contact force is not possible, as the contact force is negatively correlated with energy consumption. To reduce contact force, the leg must decelerate at the end of the swing phase, and to maintain forward speed, it must accelerate faster during the early swing. Both phases consume motor torque. Conversely, letting the leg drop almost in free fall saves torque, but the impact force spikes sharply. A straight-leg posture directs the load through the skeleton, saving holding torque and energy; however, the joints cannot absorb the impact, transmitting it directly to the ground. Conversely, a bent-knee posture allows the joints to absorb the shock, but continuously requires holding torque to support the body weight. In consequence, constrained marginal risk-sensitive RL algorithms lead to an inefficient trade-off with the desired reward return.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

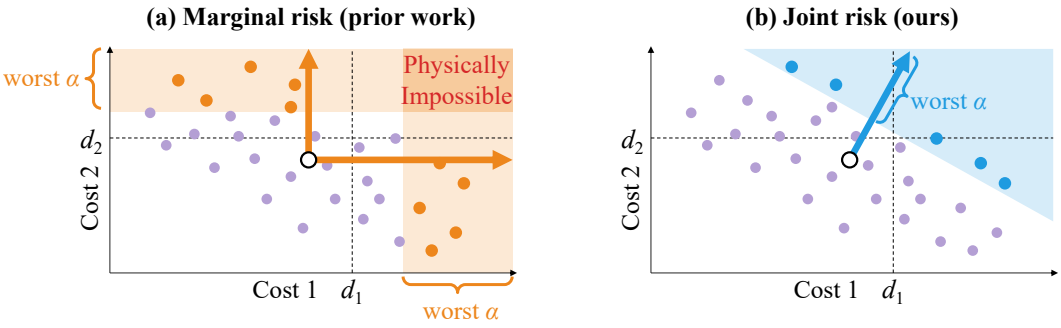

Figure 1: Marginal versus joint risk for the same policy. (a) Constraining each cost independently via marginal distributions assumes that all costs simultaneously attain their worst-case outcomes; a physically impossible event. (b) JRCPO constrains a joint risk along the reward-cost trade-off direction, where the corresponding adverse event actually occurs.

To resolve this issue, the initial step is selecting an appropriate multivariate risk, a concept for which no single consensus exists in the literature. In this work, we interpret risk as the expectation over plausible alternative scenarios yielding adverse outcomes for the RL agent. Induced by coherent duality (Ang et al., 2018), this interpretation makes the risk computable in multivariate setting, thereby facilitating the learning process of the RL agent.

Here, we propose Joint Risk Constrained Policy Optimization (JRCPO). JRCPO minimizes the multivariate CVaR, computing it through a Gaussian approximation. Through this, it manages a single risk in the direction where the policy is actually updated. We provide the justification for this, and the solution is intuitively a projection of the return in the Lagrangian direction. As a result, unlike existing marginal risk-sensitive approaches, JRCPO only accounts for realizable events. Figure 1 depicts the conceptual difference between independent marginal risk constraints and our joint-risk formulation. To the best of our knowledge, this is the first method to consider risk by utilizing the correlation among reward and multiple costs. We have successfully conducted experiments on various tasks, deriving the efficient reward-cost trade-off. Our contributions are as follows:

- We formulate a joint-risk constraint problem, which focuses risk evaluation on realizable events, avoiding unnecessary conservatism.
- We propose JRCPO, an actor-critic algorithm that evaluates this joint risk tractably under a Gaussian assumption.
- On multi-cost tasks, JRCPO satisfies all mean-cost limits and achieves higher reward than methods with marginal risk constraints.

2 Related Work

Constrained RL. Constrained RL learns a policy that maximizes the expected return subject to constraints on the expected cost return (Altman, 1999). These approaches can be broadly categorized into two streams: parameter-level and objective-level formulations. In this paper, we primarily focus on the latter. At the parameter level, CPO incorporates cost constraints into trust-region policy updates (Achiam et al., 2017). To mitigate the heavy computational cost of Hessian evaluation required by such methods, first-order approximation techniques like FOCOPS (Zhang et al., 2020) and CUP (Yang et al., 2022) have been proposed. On the other hand, objective-level approaches directly formulate the Lagrangian of the objective and solve it via dual ascent. In deep RL, PPO-Lagrangian (Ray et al., 2019) and SAC-Lagrangian (Ha et al., 2020) integrate safety constraints and multiplier updates into the policy learning (e.g., PPO or SAC). PID-Lagrangian analyzes these updates from a feedback control perspective and employs PID control to mitigate oscillations of Lagrangian multipliers during training and overshoots in constraint violations (Stooke et al., 2020).

Risk-sensitive RL. Risk-sensitive RL learns a policy to account for unfavorable outcomes in the return distribution. Tamar et al. (2015) extended policy gradients to coherent risk measures, including

CVaR. IQN models the return distribution as a quantile functions and controls the risk of low returns using distortion risk measures (Dabney et al., 2018). Combining risk-sensitive and constrained RL, WCSAC approximates the cost return distribution and imposing a CVaR constraint (Yang et al., 2023). Meanwhile, SDAC handles multiple risk constraints simultaneously through trust-region policy updates (Kim et al., 2023). SRCPO generalizes constraints to spectral risk measures, which include CVaR, and proposes a bilevel optimization framework based on their dual representation (Kim et al., 2024). None of these approaches, unlike ours, address minimizing cost violations through joint distribution-aware risk.

Multivariate risk. There exist various types of multivariate risks. Since $\mathbb{R}^n$ ($n \geq 2$) lacks a natural ordering, vector-valued or set-valued risks can be formulated by endowing it with a partial order via a convex cone (Jouini et al., 2004; Hamel et al., 2007; Abdous & Theodorescu, 1992). Ang et al. (2018) generalized the coherent duality-based formulation to the multivariate setting; this can be viewed as a type of spatial risk incorporating relative positions among realized random vectors. In that we assume a Gaussian distribution and calculate risk via projection, our approach can also be interpreted from a projected quantile perspective (Kong & Mizera, 2012). Regarding risk-sensitive RL that accounts for joint distributions, Yoo et al. (2026) proposed a risk-sensitive multi-objective RL method; however, no existing technique has been developed for constrained RL.

3 Preliminaries and Problem Formulation

3.1 Constrained Markov Decision Processes

Consider a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \mathcal{P}, \gamma)$. Here, $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $R : \mathcal{S} \times \mathcal{A} \to \triangle(\mathbb{R})$ is the reward, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \triangle(\mathcal{S})$ is the transition probability, and finally $\gamma \in [0, 1)$ is a discount factor. Here, $\triangle(X)$ is the set of probability measures on some set X. The constrained MDP (CMDP) is an augmented MDP with costs $\{C^{(i)} | C^{(i)} : \mathcal{S} \times \mathcal{A} \to \triangle(\mathbb{R}_{\geq 0})\}_{i=1}^n$, and limits $\{d_i > 0\}_{i=1}^n$, where $n \in \mathbb{N}$ is the number of constrained costs.

Let $Z_R^\pi(s_t, a_t)$ be a random variable representing the reward return of policy $\pi : \mathcal{S} \to \triangle(\mathcal{A})$ given the state-action pair s_t, a_t; i.e.,

$$Z_R^\pi(s_t, a_t) = \sum_{k=t}^{\infty} \gamma^{k-t} R_k | s_t, a_t. \tag{1}$$

Similarly, we define the cost returns $Z_{C^{(i)}}^\pi$. We denote the expected reward by $\mu_R^\pi = \mathbb{E}[Z_R^\pi]$, the cumulative cost random vector by $\mathbf{Z}_C^\pi = [Z_{C^{(1)}}^\pi, \ldots, Z_{C^{(n)}}^\pi]$ with mean $\boldsymbol{\mu}_C^\pi = \mathbb{E}[\mathbf{Z}_C^\pi]$, and the joint return by

$$\mathbf{Z}^\pi = [-Z_R^\pi, \mathbf{Z}_C^\pi] \text{ with mean } \boldsymbol{\mu}^\pi = \mathbb{E}[\mathbf{Z}^\pi] = [-\mu_R^\pi, \boldsymbol{\mu}_C^\pi]. \tag{2}$$

For simplicity, we slightly abuse notation by writing it without explicit transposes.

The objective of traditional Constrained RL is finding out optimal policy π^* such that

$$\pi^* = \arg\max_\pi \ \mu_R^\pi \ \text{ constrained to } \ \boldsymbol{\mu}_C^\pi \preceq \mathbf{d}. \tag{3}$$

Here $\preceq$ denotes the elementwise inequality and $\mathbf{d} = [d_1, \ldots, d_n]$. By introducing the Lagrangian of equation 3 with Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$, we have

$$\pi^* = \arg\max_\pi \min_{\boldsymbol{\lambda} \succeq \mathbf{0}} \left[\mathcal{L}(\pi, \boldsymbol{\lambda}) = \mu_R^\pi - \boldsymbol{\lambda}^T (\boldsymbol{\mu}_C^\pi - \mathbf{d}) \right]. \tag{4}$$

3.2 Risk

Risk is a real number associated with the severity of an outcome, and a risk measure is a functional that maps a random variable to a risk value. One of the most popular risk measures is CVaR with confidence $\alpha \in [0, 1]$, which is defined as

$$\text{CVaR}_\alpha(Z) = \frac{1}{\alpha} \int_{1-\alpha}^1 \mathbb{I}(u > 1 - \alpha) F_Z^{-1}(u) du, \tag{5}$$

where $F_Z^{-1} : [0, 1] \to \mathbb{R}$ is the quantile function of the Z.

Theorem 1 (Coherent Duality of CVaR) *The following holds:*

$$\mathrm{CVaR}_\alpha(X) = \sup_{\mathbb{Q}\in\mathcal{T}} \mathbb{E}_\mathbb{Q}[X], \tag{6}$$

where $\mathbb{P}$ is the probability measure associated with X,$\mathcal{T} = \{\mathbb{Q} \mid \frac{d\mathbb{Q}}{d\mathbb{P}} \leq \frac{1}{\alpha}$ and $\mathbb{Q} \ll \mathbb{P}\}$, and $\frac{d\mathbb{Q}}{d\mathbb{P}}$ is the Radon-Nikodym derivative[1] of some probability measure $\mathbb{Q}$ with respect to $\mathbb{P}$.

The conventional objective of constrained marginal risk-sensitive policy (Yang et al., 2023; Kim et al., 2023) is finding out π^* such that

$$\pi^* = \arg\max_\pi \ \mu_R^\pi \ \text{ constrained to } \ \mathrm{CVaR}_\alpha\left(Z_{C^{(i)}}^\pi\right) \leq d_i, \ \ i = 1, \ldots, n. \tag{7}$$

3.3 Problem Formulation

The problem formulation of conventional marginal risk-sensitive policies includes probability zero events (Ararat & Feinstein, 2024); physically impossible events. To consider the risk that is admissible for the joint distribution of $\mathbf{Z}^\pi$, we define the uncertainty set based on the coherent duality in Theorem 1 as $\mathcal{T} = \left\{\mathbb{Q} \mid \frac{d\mathbb{Q}}{d\mathbb{P}} \leq \frac{1}{\alpha} \text{ and } \mathbb{Q} \ll \mathbb{P}\right\}$, where $\mathbb{P}$ is the probability measure associated with $\mathbf{Z}^\pi$. This approach aligns with Ang et al. (2018).

Therefore, the corresponding optimization problem is formulated as:

$$\pi^* = \arg\max_\pi \mathbb{E}_\mathbb{Q}[Z_R^\pi] \ \text{ constrained to } \ \mathbb{E}_\mathbb{Q}\left[(Z_{C^{(i)}}^\pi - d_i)\right] \leq 0 \ \text{ for all } \ \mathbb{Q} \in \mathcal{T}. \tag{8}$$

The Lagrangian of equation 8 $\mathfrak{L}$ is induced as

$$\mathfrak{L}(\pi, \boldsymbol{\lambda}) = \mathbb{E}_\mathbb{Q}\left[Z_R^\pi - \sum_{i=1}^n \lambda_i (Z_{C^{(i)}}^\pi - d_i)\right] \ \text{ for all } \mathbb{Q} \in \mathcal{T} \tag{9}$$

To address the for all condition, we take the $\inf$ in equation 8, resulting

$$\begin{aligned} \inf_{\mathbb{Q}\in\mathcal{T}} \mathbb{E}_\mathbb{Q}\left[Z_R^\pi - \sum_{i=1}^n \lambda_i \left(Z_{C^{(i)}}^\pi - d_i\right)\right] &= -\sup_{\mathbb{Q}\in\mathcal{T}} \mathbb{E}_\mathbb{Q}\left[-Z_R^\pi + \sum_{i=1}^n \lambda_i \left(Z_{C^{(i)}}^\pi - d_i\right)\right] \\ &= -\underbrace{\mathrm{CVaR}_\alpha}_{\sup_{\mathbb{Q}\in\mathcal{T}} \mathbb{E}_\mathbb{Q}} \left(-Z_R^\pi + \underbrace{\boldsymbol{\lambda}^T(\mathbf{Z}_C^\pi - \mathbf{d})}_{\sum_i \lambda_i (Z_{C^{(i)}}^\pi - d_i)}\right). \end{aligned} \tag{10}$$

Since the optimal policy π^* is given by

$$\pi^* = \arg\max_\pi \min_{\boldsymbol{\lambda}\succeq\mathbf{0}} \mathfrak{L}(\pi, \boldsymbol{\lambda}) = \arg\max_\pi -\mathrm{CVaR}_\alpha\left(\frac{1}{1 + \|\boldsymbol{\lambda}^*\|_1}\left(-Z_R^\pi + \boldsymbol{\lambda}^{*T}(\mathbf{Z}_C^\pi - \mathbf{d})\right)\right), \tag{11}$$

where $\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}\succeq\mathbf{0}} \mathfrak{L}(\pi, \boldsymbol{\lambda})$, we consider the normalized Lagrange multiplier.

4 Approach

Calculating the CVaR_α for the general distribution equation 10 is non-trivial, requiring a minimax problem for each data point (Kong & Mizera, 2012). To address this, we show that in a repetitive environment (i.e., ergodic), it is possible to approximate $\mathbf{Z}^\pi$ by a Gaussian distribution. Due to the property that any projection of a Gaussian distribution remains Gaussian, CVaR_α can be easily calculated. Furthermore, we show that the approximation is practical in non-ergodic environments empirically in Section 5.

Figure 2 describes the JRCPO framework. First, the critic takes the state s and the action $a \sim \pi(\cdot|s)$ as input, then it outputs the mean $\boldsymbol{\mu}$ and the covariance Σ of the joint return, learned in normalized

[1] The ratio of two probability density functions.

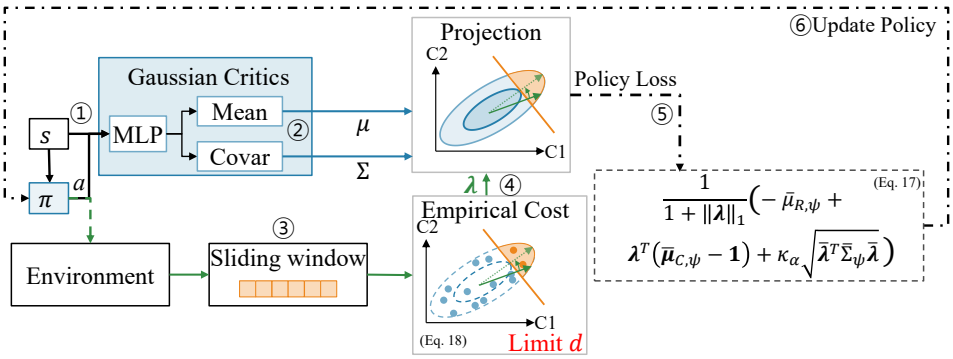

Figure 2: Overview of JRCPO. Circled numbers show the policy update process: the critic estimates joint-return distribution, empirical returns update the Lagrange multipliers, and the multipliers set the direction of joint risk.

scale. Then, the Lagrange multipliers $\boldsymbol{\lambda}$, updated according to the mean and the covariance of empirical sliding window, set the direction of the joint risk. Next, CVaR is calculated with Theorem 3. The mean reward, costs and the joint risk constitute the policy loss in equation 13, and finally the actor is updated to minimize this loss. The following sections describe policy loss from the Gaussian approximation, normalized critic, and dual update.

4.1 Gaussian Approximation of Return Distribution

The following theorem provides the justification for the Gaussian return approximation and we can obtain the actor loss, which is aligned with equation 10.

Theorem 2 *(Srikant, 2025) Let $(\mathcal{S}, \mathcal{A}, \mathbf{R}, \mathcal{P}, \gamma)$ be a multi-objective MDP whose transition is ergodic. That is, $\Pr[s_{t+k} = s' \mid s, \pi] > 0$ for any $s' \in \mathcal{S}$ for a sufficiently large $k \in \mathbb{N}$. Suppose further that π is a stationary policy. Then, for a sufficiently large $\gamma \in (0, 1)$, the policy return $\mathbf{Z}_R^\pi$, viewed as a random vector, can be approximated by a Gaussian distribution. This characterizes the return distribution of multi-objective MDPs, it directly applies to CMDPs.*

Theorem 3 *Let $\boldsymbol{\lambda} \in \mathbb{R}^n$ be a constant vector. Let $\mathbf{Z}$ be a random vector following a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the mean vector and $\Sigma \in M(\mathbb{R})_{n\times n}$ is the covariance matrix. Then,*

$$\mathrm{CVaR}_\alpha(\boldsymbol{\lambda}^T\mathbf{Z}) = \boldsymbol{\lambda}^T\boldsymbol{\mu} + \kappa_\alpha\sqrt{\boldsymbol{\lambda}^T\Sigma\boldsymbol{\lambda}}, \tag{12}$$

where $\kappa_\alpha = \frac{1}{\alpha}\phi(\Phi^{-1}(\alpha))$. Here, ϕ and Φ^{-1} are the density and quantile functions of $\mathcal{N}(0, 1)$.

The proof follows directly from the results of Rockafellar & Uryasev (2000); $\mathrm{CVaR}_\alpha(X)$ of $X \sim \mathcal{N}(\mu, \sigma^2)$ is $\mu + \kappa_\alpha\sigma$, and $\boldsymbol{\lambda}^T\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\lambda}^T\boldsymbol{\mu}, \boldsymbol{\lambda}^T\Sigma\boldsymbol{\lambda})$.

By solving equation 10 using Theorems 2 and 3, we obtain the following result:

$$\mathcal{L}(\pi) = \frac{1}{1 + \|\boldsymbol{\lambda}\|_1}\left(-\mu_R^\pi + \boldsymbol{\lambda}^T(\boldsymbol{\mu}_C^\pi - \mathbf{d}) + \kappa_\alpha\sqrt{\bar{\boldsymbol{\lambda}}^T\Sigma\bar{\boldsymbol{\lambda}}}\right). \tag{13}$$

where $\Sigma = \mathbb{C}\mathrm{ov}[\mathbf{Z}^\pi]$ and $\bar{\boldsymbol{\lambda}} = [1, \lambda_1, \ldots, \lambda_n]$.

4.2 Normalized Joint Gaussian Critic

Unlike existing approaches, JRCPO directly models the joint return vector $\mathbf{Z}^\pi(s, a)$ under a Gaussian approximation. Let $\boldsymbol{b}$ be the mean of the Bellman target $[-R, \mathbf{C}] + \gamma\mathbf{Z}_{t+1}^\pi$, and let B be its covariance matrix. Define the mean Bellman residual as $\boldsymbol{\varepsilon} := \boldsymbol{\mu}^\pi - \boldsymbol{b}$. Then, the gradient of the Gaussian negative log-likelihood is given by

$$\nabla_{\boldsymbol{\mu}}\mathcal{L}_{\mathrm{NLL}} = \Sigma^{-1}\boldsymbol{\varepsilon}, \qquad \nabla_\Sigma\mathcal{L}_{\mathrm{NLL}} = \frac{1}{2}\Sigma^{-1}\left(\Sigma - B - \boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right)\Sigma^{-1}. \tag{14}$$

Algorithm 1 JRCPO

Require: reward scale s_R, limits $\mathbf{d}$, CVaR confidence α
Initialize actor π_θ, critic $(\bar{\boldsymbol{\mu}}_\psi, L_\psi)$, replay buffer $\mathcal{D}$, sliding window $W \leftarrow \emptyset$, and $\boldsymbol{\lambda} \leftarrow \mathbf{0}$
for each iteration **do**
  for each environment step **do**
    $a \sim \pi_\theta(\cdot \mid s)$, observe $(r, \mathbf{c}, s')$, and add $(s, a, r, \mathbf{c}, s')$ to $\mathcal{D}$
    if episode terminates **then**
      Add $\mathbf{Z}^\pi_{\text{samples}}$ to W
    end if
  end for
  Update $\boldsymbol{\lambda}$ using $\widehat{\boldsymbol{\mu}}^\pi_C$ and $\widehat{\Sigma}$ computed from W ▷ equation 18
  for each gradient step **do**
    Sample a minibatch from $\mathcal{D}$
    Update the critic by minimizing the Gaussian NLL of $\bar{\mathbf{Z}}^\pi$ ▷ equation 16
    Update the actor by minimizing $\mathcal{L}(\pi_\theta)$ ▷ equation 17
  end for
end for

The outer product $\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T$ increases quadratically with $\boldsymbol{\varepsilon}$, severely distorting the scale. Under inverse-covariance weighting (Seitzer et al., 2022), the covariance increases to fit this distortion, which subsequently decreases the gradient of the mean. Moreover, Theorem 2 requires a sufficiently large γ, which results in a long effective horizon of approximately $\frac{1}{1-\gamma}$ steps. Over such a long horizon, small per-step reward and cost differences accumulate into scale gaps; thus, normalization is essential to prevent large scale components from dominating the joint critic updates.

Because each limit d_i reflects the scale of both immediate and cumulative costs, it serves as a natural baseline for normalization. Accordingly, we normalize each cost dimension in $\mathbf{Z}^\pi(s,a)$ by d_i, and the reward by a task-level scale $s_R > 0$. This scale s_R is the sole additional hyperparameter in JRCPO; we set it directly using the return of an unconstrained reward-only policy without tuning (Appendix A.2). Formally, we construct a diagonal scaling matrix $\mathbf{N}$ to define the normalized return, mean, and covariance as follows,

$$\bar{\mathbf{Z}}^\pi(s,a) = \mathbf{N}^{-1}\mathbf{Z}^\pi(s,a),\ \mathbf{N} = \mathrm{diag}(s_R, d_1, \ldots, d_n),\ \bar{\boldsymbol{\mu}}^\pi = \mathbf{N}^{-1}\boldsymbol{\mu}^\pi,\ \bar{\Sigma} = \mathbf{N}^{-1}\Sigma\mathbf{N}^{-1}. \quad (15)$$

Here $\boldsymbol{\mu}^\pi$ and $\mathbf{Z}^\pi$ are defined in equation 2. To estimate the joint mean and covariance applied in equation 13, we parameterize the critic by ψ as follows:

$$\bar{\mathbf{Z}}^\pi(s,a) \approx \mathcal{N}\left(\bar{\boldsymbol{\mu}}_\psi(s,a), \bar{\Sigma}_\psi(s,a)\right), \qquad \bar{\Sigma}_\psi = L_\psi L_\psi^T, \quad (16)$$

where L_ψ is a lower-triangular Cholesky factor. We also include an entropy bonus in the reward return (Haarnoja et al., 2018). This is because a deterministic optimal policy may not exist in CMDPs (Kalagarla et al., 2025).

With this normalization applied, the final actor loss is defined as:

$$\mathcal{L}(\pi) = \frac{1}{1 + \|\boldsymbol{\lambda}\|_1}\left(-\bar{\mu}_{R,\psi} + \boldsymbol{\lambda}^T(\bar{\boldsymbol{\mu}}_{C,\psi} - \mathbf{1}) + \kappa_\alpha\sqrt{\bar{\boldsymbol{\lambda}}^T\bar{\Sigma}_\psi\bar{\boldsymbol{\lambda}}}\right), \quad (17)$$

where $\mathbf{1} = [1, \ldots, 1]$.

4.3 Dual Update

The Lagrange multiplier is updated based on the empirical cost (Stooke et al., 2020). However, covariance cannot be computed using a single sample. To address this, we accumulate a history of empirical joint returns using a sliding window W for the update.

Critic estimates may underestimate true episodic cost returns, causing critic-based multiplier updates to miss constraint violations (Hogewind et al., 2023; Wu et al., 2024). Therefore, we update $\boldsymbol{\lambda}$ using empirical returns from the completed episodes' results. For each episode, we collect the discounted, normalized joint return sample $\mathbf{Z}^\pi_{\text{sample}}$ and calculate normalized empirical cost mean $\widehat{\boldsymbol{\mu}}^\pi_C$ and covariance $\widehat{\Sigma}$ from the W.

Table 1: Main results under stochastic policies. Each cost row gives the component name and its limit. All quantities are discounted and in original task units.

	SAC-Lag	PPO-Lag	SAC-PID	SDAC	SRCPO	WCSAC	**JRCPO** (Ours)
PointCircle1							
Return	0.39 ±11.70	28.54 ±4.56	0.20 ±4.75	7.88 ±6.32	6.50 ±1.32	1.11 ±1.93	20.47 ±0.36
boundary (≤ 25)	64.4 ±81.4 / 165	26.2 ±3.84 / 102	89.9 ±58.8 / 297	56.3 ±108 / 102	7.22 ±4.06 / 66.1	22.6 ±32.6 / 120	22.9 ±7.57 / 87.4
drive (≤ 133)	221 ±77.1 / 302	100 ±22.8 / 120	168 ±61.5 / 249	110 ±158 / 133	47.2 ±15.3 / 101	230 ±86.2 / 358	115 ±1.05 / 124
turn (≤ 53)	113 ±44.0 / 206	52.7 ±2.34 / 67.7	106 ±39.2 / 197	89.8 ±170 / 93.7	24.8 ±7.52 / 43.1	130 ±111 / 253	50.1 ±0.702 / 57.8
CarGoal2							
Return	−0.64 ±0.14	4.43 ±1.19	−0.72 ±0.16	7.97 ±0.99	7.30 ±1.51	4.33 ±5.91	7.98 ±0.70
hazards (≤ 6)	4.15 ±2.38 / 29.1	5.09 ±0.986 / 26.8	6.08 ±0.473 / 38.7	4.18 ±0.587 / 22.2	3.47 ±1.04 / 19.1	27.3 ±11.0 / 139	5.49 ±2.12 / 35.2
vase contact (≤ 6)	2.84 ±0.995 / 20.5	4.01 ±0.839 / 22.8	3.52 ±0.432 / 24.2	1.72 ±0.410 / 10.7	2.27 ±0.420 / 13.7	13.4 ±8.70 / 49.1	1.54 ±0.189 / 8.54
vase velocity (≤ 19)	14.9 ±4.12 / 90.6	18.0 ±2.66 / 89.4	18.1 ±1.81 / 105	9.43 ±2.36 / 42.7	10.9 ±1.57 / 50.5	49.9 ±25.9 / 159	12.0 ±1.91 / 54.3
HalfCheetahVelocity							
Return	680 ±1048	1978 ±3	1367 ±798	886 ±803	1576 ±92	120 ±1131	1913 ±9
velocity (≤ 25)	12.1 ±16.5 / 15.8	29.0 ±2.53 / 37.2	53.1 ±83.4 / 83.3	0.0733 ±0.164 / 0.290	4.84 ±3.18 / 14.1	7.09 ±15.9 / 9.02	22.8 ±7.26 / 29.9
pitch (≤ 44)	259 ±490 / 386	31.2 ±5.27 / 37.0	332 ±609 / 401	180 ±234 / 183	15.8 ±10.8 / 51.7	70.1 ±101 / 200	28.9 ±3.47 / 33.0
control (≤ 127)	170 ±114 / 175	123 ±13.2 / 124	184 ±96.2 / 189	73.6 ±7.50 / 75.2	72.2 ±2.94 / 73.5	92.9 ±61.9 / 115	87.9 ±4.30 / 90.6
HopperVelocity							
Return	751 ±433	558 ±447	849 ±326	378 ±228	249 ±251	26 ±24	722 ±237
velocity (≤ 25)	147 ±193 / 248	15.5 ±8.51 / 35.8	198 ±182 / 352	3.83 ±7.58 / 6.23	1.32 ±2.86 / 1.94	8.42 ±7.71 / 8.94	23.3 ±33.1 / 71.2
landing load (≤ 16)	22.2 ±14.1 / 38.9	9.13 ±9.14 / 10.2	29.3 ±8.65 / 41.4	4.34 ±2.24 / 4.79	3.65 ±2.57 / 6.01	1.35 ±0.388 / 1.44	11.6 ±5.47 / 16.7
control (≤ 41)	78.8 ±24.1 / 103	19.9 ±14.9 / 24.0	95.9 ±5.23 / 113	24.3 ±12.2 / 32.6	29.3 ±30.0 / 40.4	5.54 ±1.71 / 5.85	31.6 ±10.0 / 48.3
MO-LunarLander							
Return	−104.72 ±195.35	124.77 ±48.88	−181.39 ±291.06	69.79 ±27.03	20.53 ±11.83	−167.69 ±205.82	263.92 ±4.69
main engine (≤ 92)	275 ±155 / 309	48.3 ±19.0 / 96.8	251 ±106 / 339	41.5 ±14.0 / 85.7	13.2 ±9.52 / 40.7	168 ±61.1 / 318	71.8 ±8.31 / 100
side engine (≤ 27)	49.3 ±27.0 / 74.4	15.1 ±3.16 / 43.6	70.2 ±29.4 / 139	10.9 ±2.43 / 32.4	6.60 ±1.81 / 19.0	35.6 ±24.5 / 65.6	16.1 ±1.56 / 38.6

The sliding window W smooths isolated noise while tracking sustained distribution shifts (Chen et al., 2026). Similarly to equation 17, the cost limits reduce to $\mathbf{1}$. Accordingly, the multipliers are updated via projected gradient ascent using empirical constraint violations evaluated with fixed $\boldsymbol{\lambda}_t$:

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \eta_\lambda \left(\widehat{\boldsymbol{\mu}}_C^{\pi} - \mathbf{1} + \kappa_\alpha \sqrt{\bar{\boldsymbol{\lambda}}_t^T \widehat{\Sigma} \bar{\boldsymbol{\lambda}}_t \mathbf{1}}\right)\right]_+, \tag{18}$$

where η_λ denotes the learning rate of the Lagrange multiplier.

5 Experiments

5.1 Experimental Setup

Tasks. We evaluate multi-cost formulations of five continuous-control tasks: PointCircle1, CarGoal2, HalfCheetahVelocity, and HopperVelocity from Safety-Gymnasium (Ji et al., 2023), and MO-LunarLander, a non-ergodic environment adapted from MO-Gymnasium. We additionally consider an EV2Gym charging case study (Orfanoudakis et al., 2025) in which a policy maximizes energy-trading profit subject to a departure-shortfall constraint under uncertainty early departures.

Baseline Algorithms. We compare JRCPO with three objective-level, expected-cost Lagrangian methods: SAC-Lag (Ha et al., 2020), PPO-Lag (Ray et al., 2019), and SAC-PID (Stooke et al., 2020). We also compare JRCPO with three marginal risk-sensitive methods: SDAC (Kim et al., 2023), SRCPO (Kim et al., 2024), and WCSAC(Yang et al., 2023), . For EV2Gym, we additionally include the as-fast-as-possible (AFAP) charging heuristic, which is the lowest-cost baseline.

Training and Evaluation. Every method is trained with five seeds in each task, and each trained policy is evaluated for 1000 episodes. Because optimal CMDP policies may require stochastic property (Altman, 1999; Agarwal et al., 2022), we evaluate every learned method by stochastic policy. Across all evaluation results, gray shading denotes entries that violate the cost limit, and values report the mean ± standard deviation across random seeds ($X \pm Y$).

Appendix A.1 provides detailed tasks and Appendix A.3 provides the detailed implementations.

5.2 Main Results

JRCPO satisfies every mean-cost limit in all five multi-cost tasks and achieves a higher mean return on each task than SRCPO, the only baseline with the same feasibility coverage. In PointCircle1 and

Method	$\Delta_{\max} = 0$ Return ↑	$\Delta_{\max} = 0$ $C \downarrow$	$\Delta_{\max} = 4$ Return ↑	$\Delta_{\max} = 4$ $C \downarrow$	$\Delta_{\max} = 8$ Return ↑	$\Delta_{\max} = 8$ $C \downarrow$
EV2Gym		Limit: 0.1		Limit: 0.1		Limit: 0.1
AFAP	−14.17	0 / 0	−14.17	0 / 0	−14.17	0 / 0
SAC-Lag	5.01 ±2.58	0.129 ±0.0159 / 0.498	4.08 ±2.48	0.125 ±0.0165 / 0.483	3.21 ±2.46	0.122 ±0.0168 / 0.465
PPO-Lag	72.44 ±0.32	0.782 ±0.00222 / 0.828	71.08 ±0.22	0.774 ±0.00343 / 0.824	68.81 ±0.16	0.757 ±0.00432 / 0.817
SAC-PID	63.25 ±0.39	0.652 ±0.00639 / 0.770	60.31 ±0.43	0.629 ±0.00669 / 0.756	56.88 ±0.48	0.603 ±0.00696 / 0.741
SDAC	8.05 ±1.56	0.116 ±0.0125 / 0.186	3.90 ±1.23	0.0985 ±0.0100 / 0.168	−0.52 ±0.95	0.0833 ±0.00786 / 0.153
SRCPO	1.25 ±1.19	0.0621 ±0.00420 / 0.134	1.14 ±1.12	0.0651 ±0.00469 / 0.138	0.59 ±1.10	0.0676 ±0.00559 / 0.144
WCSAC	23.69 ±4.15	0.164 ±0.0196 / 0.708	23.08 ±3.65	0.163 ±0.0182 / 0.695	20.75 ±3.40	0.158 ±0.0171 / 0.666
JRCPO	5.35 ±1.73	0.0730 ±0.00600 / 0.430	5.88 ±1.69	0.0774 ±0.00545 / 0.421	5.66 ±1.49	0.0817 ±0.00414 / 0.403

Table 2: EV2Gym results for maximum early-departure shifts $\Delta_{\max}$ (steps). AFAP has no across-run SD. Here $\Delta_{\max}$ is the maximum early-departure shift in steps; policies trained at $\Delta_{\max} = 4$ are evaluated under both milder and more severe departure uncertainty.

HalfCheetahVelocity, PPO-Lag achieves higher returns but exceeds a mean-cost limit, while SRCPO leaves more each limit unused and achieves lower returns than JRCPO. On MO-LunarLander, JRCPO achieves the highest mean return among methods satisfying all mean-cost limits, showing the applicability of JRCPO in a non-ergodic environment. Table 1 summarizes the main results.

In Ev2Gym environment, JRCPO and SRCPO are the only learned methods that satisfy the mean departure-shortfall limit across all three early-departure settings. JRCPO shows robustness to the environment's dynamics, achieving the highest profit under large maximum early departure shifts. Here, AFAP serves as a baseline for the lowest profit. Table 2 depicts the results.

5.3 Ablation Studies

In this subsection, along with an ablation study on each component, we conduct an analysis of whether JRCPO effectively mitigates risk in the direction of the Lagrange multiplier.

Joint-risk components. Table 3 compares three risk variants on MO-LunarLander and HalfCheetahVelocity. *Diagonal* uses only marginal variances in risk evaluation and *Cost-only* removes reward from the Lagrangian multiplier's direction. The results on MO-LunarLander demonstrate that every component is essential. Specifically, the outcomes from the *Diagonal* variant on HalfCheetah and the *Risk-neutral* variant on MO-LunarLander reveal that omitting them impairs numerical stability.

Reward normalization and scale sensitivity. *Half*, *Double*, and *Unscaled* denote setting s_R to $s_R/2$, $2s_R$, and 1, respectively. *Unscaled* retains cost normalization but lowers mean return and violates mean-cost limits on both tasks. Although *Half* and *Double* violate the cost limit, the violation ratio is below 50%, indicating that the method is not overly sensitive to s_R.

Table 3: JRCPO ablation results. Metrics follow Table 1. Results are averaged over five runs, except † (two valid runs) and ‡ (four valid runs), with numerical failures. **Bold Gray** denotes cases where the cost violation is less than 50%.

Component	Variant	MO-LunarLander Return ↑	MO-LunarLander $C_1 \downarrow$ Limit: 92	MO-LunarLander $C_2 \downarrow$ Limit: 27	HalfCheetahVelocity Return ↑	HalfCheetahVelocity $C_1 \downarrow$ Limit: 25	HalfCheetahVelocity $C_2 \downarrow$ Limit: 44	HalfCheetahVelocity $C_3 \downarrow$ Limit: 127
JRCPO (Ours)		263.92 ±4.69	71.8 ±8.31 / 100	16.1 ±1.56 / 38.6	1913 ±9	22.8 ±7.26 / 29.9	28.9 ±3.47 / 33.0	87.9 ±4.30 / 90.6
Joint risk	Diagonal	206.52 ±49.06	212 ±146 / 264	24.8 ±12.6 / 43.9	1907‡ ±14	17.4 ±4.69 / 23.6	34.1 ±3.68 / 41.1	99.8 ±10.3 / 103
	Cost-only	143.39 ±32.31	357 ±10.1 / 376	**30.2** ±10.0 / 47.4	1904 ±19	20.2 ±6.38 / 27.0	30.7 ±3.41 / 43.8	90.4 ±11.3 / 94.9
	Risk-neutral	−1637.29† ±1493.47	395 ±5.49 / 428	237 ±100 / 336	386 ±588	18.5 ±41.4 / 24.6	198 ±273 / 446	291 ±31.0 / 321
Reward scale	Half	**245.89** ±48.15	**132** ±129 / 157	17.3 ±5.35 / 31.5	**1924** ±5	**36.9** ±4.46 / 45.8	30.5 ±2.65 / 35.1	82.6 ±8.56 / 85.4
	Double	**141.99** ±138.24	**127** ±133 / 154	17.0 ±3.44 / 30.1	1887 ±21	20.4 ±9.51 / 27.2	33.3 ±11.0 / 73.0	97.3 ±7.67 / 107
	Unscaled	172.32 ±58.11	309 ±120 / 330	57.9 ±28.8 / 76.4	**1766** ±97	10.8 ±11.6 / 15.8	**52.6** ±16.4 / 61.6	**160** ±28.4 / 166

Analysis of risk reduction along the multiplier direction. In Table 4, JRCPO shows the lowest dispersion along the Lagrange multiplier direction in MO-LunarLander. Although JRCPO stays closer to the cost limit than the other two methods in Table 1, it achieves the smallest dispersion

along this direction. To quantify this dispersion, we define:

$$\text{dispersion}(\mathbf{Z}^\pi) := \frac{1}{1+\|\boldsymbol{\lambda}\|_1}\left(\text{CVaR}_\alpha\left(\bar{\boldsymbol{\lambda}}^T\bar{\mathbf{Z}}^\pi\right) - \mathbb{E}\left[\bar{\boldsymbol{\lambda}}^T\bar{\mathbf{Z}}^\pi\right]\right), \tag{19}$$

where $\bar{\mathbf{Z}}^\pi$ is defined in equation 15 and $\bar{\boldsymbol{\lambda}}$ is defined in equation 13.

Among the methods satisfying the cost limits, JRCPO achieves a dispersion of 0.743, which is approximately one-fifth that of PPO-Lag (3.995) and SDAC (3.721).

Because SRCPO does not maintain a Lagrange multiplier, we report the episode distributions of both methods in Figure 3 instead of Table 4. Figure 3 shows that JRCPO tends toward higher returns; while SRCPO episodes spread widely toward low returns, JRCPO exhibits little variance along $\boldsymbol{\lambda}$ and operates closer to the cost limits. Appendix B provides the results for other tasks.

Table 4: Dispersion in equation 19 along each method's final multiplier direction on MO-LunarLander. SDAC's multipliers are estimated post hoc from its final actor and critics.

	SAC-Lag	PPO-Lag	SAC-PID	SDAC	WCSAC	**JRCPO**
dispersion	0.735 ±0.661	3.995 ±0.776	1.700 ±0.937	3.721 ±1.276	1.241 ±0.491	0.743 ±0.371

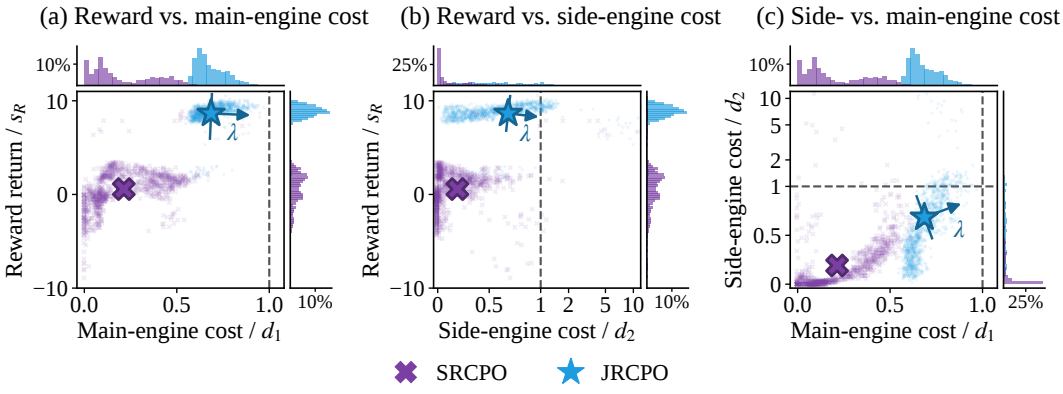

Figure 3: Joint distributions of reward and cost returns of JRCPO and SRCPO on MO-LunarLander, using one training run per method. Cost returns are normalized by their limits d_1 and d_2, and reward returns by the fixed reward scale s_R. Small points represent episodes, large markers indicate means, and marginal histograms show episode percentages. Arrows indicate pairwise projections of JRCPO's adverse direction determined by its final multipliers; short lines through its means are perpendicular visual guides in display coordinates. Side-engine cost axes use symlog scaling with linear threshold 1.

6 Conclusion and Limitation

We introduce JRCPO, a joint risk-aware constrained RL framework in reward and cost returns. We use CVaR duality to evaluate a risk along the Lagrangian direction achieving efficient reward-cost trade-off. To calculate this risk, we employ a joint Gaussian critic that captures the covariance structure among returns and enables closed-form CVaR evaluation. We also normalize reward and cost returns to learn their joint distribution on a common scale. Experiments on multi-cost tasks demonstrate that JRCPO achieves higher mean rewards than methods with cost-wise marginal risk constraints while satisfying the mean-cost limitations. Although the Gaussian assumption ensures computational tractability for the CVaR evaluation, it inevitably imposes certain structural priors on the joint distribution. Extending this framework to accommodate more flexible distributional models remains an interesting direction for future research.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation and implementing methods. We have not used generative AI tools for interpreting results, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, and designing or providing feedback on research methodology or experiments, and generating synthetic data sets, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code, summarizing and analysing existing literature, sourcing and searching for information, formatting references, and identifying relevant literature. We have reviewed all AI-assisted work. We read and revised all AI-translated text, reviewed and tested all code written or modified with AI tools for correctness, and checked all related work identified with AI tools against the original papers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Belkacem Abdous and Radu Theodorescu. Note on the spatial quantile of a random vector. *Statistics & Probability Letters*, 13(4):333–336, 1992.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017.

Mridul Agarwal, Qinbo Bai, and Vaneet Aggarwal. Regret guarantees for model-based reinforcement learning with long-term average constraints. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2022.

Eitan Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1999.

Marcus Ang, Jie Sun, and Qiang Yao. On the dual representation of coherent risk measures. *Annals of Operations Research*, 262(1):29–46, 2018.

Çağın Ararat and Zachary Feinstein. On the separability of vector-valued risk measures. *SIAM Journal on Financial Mathematics*, 15(4):SC68–SC79, 2024.

Elliot Chane-Sane, Pierre-Alexandre Leziart, Thomas Flayols, Olivier Stasse, Philippe Souères, and Nicolas Mansard. Cat: Constraints as terminations for legged locomotion reinforcement learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13303–13310, 2024. doi: 10.1109/IROS58592.2024.10802334.

Keru Chen, Honghao Wei, Zhigang Deng, and Sen Lin. Towards fast safe online reinforcement learning via policy finetuning. *Transactions on Machine Learning Research*, 2026. URL https://arxiv.org/abs/2412.04426.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pp. 1096–1105. PMLR, 2018.

Bayaraa Enkhsaikhan and Ohyun Jo. Risk-constrained reinforcement learning with augmented lagrangian multiplier for portfolio optimization. *IEEE Transactions on Big Data*, 11(5):2489–2502, 2025. doi: 10.1109/TBDATA.2025.3533905.

Sehoon Ha, Peng Xu, Zhenyu Tan, Sergey Levine, and Jie Tan. Learning to walk in the real world with minimal human effort, 2020. URL https://arxiv.org/abs/2002.08550.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Andreas Hamel, Frank Heyde, and Markus Höhne. *Set valued measures of risk*. Inst. für Mathematik, 2007.

Yannick Hogewind, Thiago D. Simão, Tal Kachman, and Nils Jansen. Safe reinforcement learning from pixels using a stochastic latent representation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.01801.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Juntao Dai, and Yaodong Yang. Safety-Gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems*, volume 36, pp. 18964–18993. Curran Associates, Inc., 2023. doi: 10.52202/075280-0831.

Elyes Jouini, Moncef Meddeb, and Nizar Touzi. Vector-valued coherent risk measures. *Finance and stochastics*, 8(4):531–552, 2004.

Krishna Chaitanya Kalagarla, Rahul Jain, and Pierluigi Nuzzo. A safe bayesian learning algorithm for constrained mdps with bounded constraint violation. In *AISTATS*, pp. 2782–2790, 2025.

Dohyeong Kim, Kyungjae Lee, and Songhwai Oh. Trust region-based safe distributional reinforcement learning for multiple constraints. In *Advances in Neural Information Processing Systems*, volume 36, pp. 19908–19939. Curran Associates, Inc., 2023. doi: 10.52202/075280-0874.

Dohyeong Kim, Taehyun Cho, Seungyub Han, Hojun Chung, Kyungjae Lee, and Songhwai Oh. Spectral-risk safe reinforcement learning with convergence guarantees. In *Advances in Neural Information Processing Systems*, volume 37, pp. 92465–92498. Curran Associates, Inc., 2024. doi: 10.52202/079017-2936.

Linglong Kong and Ivan Mizera. Quantile tomography: using quantiles with multivariate data. *Statistica Sinica*, 22:1589–1610, 2012.

Puze Liu, Davide Tateo, Haitham Bou Ammar, and Jan Peters. Robot reinforcement learning on the constraint manifold. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1357–1366. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/liu22c.html.

Stavros Orfanoudakis, Cesar Diaz-Londono, Yunus Emre Yılmaz, Peter Palensky, and Pedro P. Vergara. EV2Gym: A flexible V2G simulator for EV smart charging research and benchmarking. *IEEE Transactions on Intelligent Transportation Systems*, 26(2):2410–2421, 2025. doi: 10.1109/TITS.2024.3510945.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. OpenAI, 2019. URL https://cdn.openai.com/safexp-short.pdf.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aPOpXlnV1T.

Rayadurgam Srikant. Rates of convergence in the central limit theorem for markov chains, with an application to td learning. *Mathematics of Operations Research*, 2025.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID Lagrangian methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143. PMLR, 2020. URL https://proceedings.mlr.press/v119/stooke20a.html.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/024d7f84fff11dd7e8d9c510137a2381-Abstract.html.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6 (3):4915–4922, 2021. doi: 10.1109/LRA.2021.3070252.

Chengyu Wang, Shuang Zhou, Luhan Wang, Zhaoming Lu, Celimuge Wu, Xiangming Wen, and Guochu Shou. Autonomous driving via knowledge-enhanced safe reinforcement learning. *IEEE Transactions on Intelligent Vehicles*, 10(1):359–372, 2025. doi: 10.1109/TIV.2024.3412916.

Yi Wang, Dawei Qiu, Mingyang Sun, Goran Strbac, and Zhiwei Gao. Secure energy management of multi-energy microgrid: A physical-informed safe reinforcement learning approach. *Applied Energy*, 335:120759, 2023. doi: 10.1016/j.apenergy.2023.120759.

Zifan Wu, Bo Tang, Qian Lin, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Off-policy primal-dual safe reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2f8b56543953d60f262fb2c4b85c50b3-Abstract-Conference.html.

Long Yang, Jiaming Ji, Juntao Dai, Linrui Zhang, Binbin Zhou, Pengfei Li, Yaodong Yang, and Gang Pan. Constrained update projection approach to safe policy optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9111–9124. Curran Associates, Inc., 2022. doi: 10.52202/068431-0662. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/3ba7560b4c3e66d760fbdd472cf4a5a9-Paper-Conference.pdf.

Qisong Yang, Thiago D. Simão, Simon H. Tindemans, and Matthijs T. J. Spaan. Safety-constrained reinforcement learning with a distributional safety critic. *Machine Learning*, 112(3):859–887, 2023. doi: 10.1007/s10994-022-06187-8.

Gwangpyo Yoo, Woo Kyung Kim, and Honguk Woo. Knothe-rosenblatt quantile regression for risk-sensitive multi-objective reinforcement learning. In *Forty-third International Conference on Machine Learning*. PMLR, 2026.

Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15338–15349. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/af5d5ef24881f3c3049a7b9bfe74d58b-Paper.pdf.

A Experimental Details

A.1 Task, Reward, and Cost Definitions

PointCircle1. Let x_t denote the horizontal simulator coordinate and let $a_t = (a_{\mathrm{drive},t}, a_{\mathrm{turn},t}) \in [-1,1]^2$ contain the forward-backward drive and turning commands. The native reward encourages tangential motion around the target circle, downweighted by radial deviation. The ordered cost vector is

$$\left[\mathbf{1}\{|x_t| > x_{\max}\}, a_{\mathrm{drive},t}^2, a_{\mathrm{turn},t}^2\right]. \tag{20}$$

The boundary threshold is $x_{\max} = 1.125$, and the squared commands measure actuator usage.

CarGoal2. CarGoal2 uses the Safety-Gymnasium level-2 Goal layout with the car robot. The native reward measures progress toward the goal and rewards reaching it. Under the default indicator-cost setting, the ordered cost vector is

$$\left[c_{\mathrm{hazards},t}, c_{\mathrm{vase\text{-}contact},t}, c_{\mathrm{vase\text{-}velocity},t}\right]. \tag{21}$$

The components indicate penetration into any hazard region, contact between the robot and any vase, and the speed of any vase reaching the simulator threshold of $10^{-4}\,\mathrm{m\,s^{-1}}$, respectively. Each component is binary.

The goal can lie outside the range of the goal lidar, which makes this task partially observable. We therefore use a history variant that keeps the reward and cost definitions unchanged and modifies only the observation. The goal lidar is computed with a maximum distance of 10 instead of the upstream default of 3, and the remaining lidars keep their defaults. The observation concatenates the $H = 11$ most recent tokens, oldest first,

$$o_t^{\mathrm{hist}} = \left[\tau_{t-H+1}, \ldots, \tau_t\right], \qquad \tau_t = \left[o_t, a_{t-1}, R_{t-1}, c_{t-1}\right], \tag{22}$$

where o_t is the flattened native observation under the lidar setting above and a_{t-1}, R_{t-1}, and c_{t-1} are the action, reward, and cost vector of the previous step. The observation dimension is therefore $H(\dim o + \dim a + 1 + n)$, where n is the number of cost components. At the start of an episode, the history repeats the initial observation H times with the action, reward, and cost set to 0.

HalfCheetahVelocity. Let $v_{x,t}$ be forward velocity, $q_{\mathrm{pitch},t}$ the post-step pitch observation, and a_t the action. We use the native forward-running reward without its control penalty. The ordered cost vector is

$$\left[\mathbf{1}\{v_{x,t} > v_{\max}\}, \left|\mathrm{atan2}(\sin q_{\mathrm{pitch},t}, \cos q_{\mathrm{pitch},t})\right|, w_{\mathrm{ctrl}}\|a_t\|_2^2\right]. \tag{23}$$

The costs measure native velocity-limit violations, absolute pitch wrapped to its principal angle, and control effort, with $v_{\max} = 3.2096\,\mathrm{m\,s^{-1}}$ and $w_{\mathrm{ctrl}} = 0.1$.

HopperVelocity. Let $v_{x,t}$ be forward velocity, a_t the action, F_ℓ the normal contact force between the foot and the floor at physics substep ℓ of the action step, n_{sub} the number of physics substeps per action step, and $m|g|$ the body weight of the robot. We use the native forward and healthy rewards without the control penalty. The ordered cost vector is

$$\left[\mathbf{1}\{v_{x,t} > v_{\max}\}, \frac{w_{\mathrm{load}}}{n_{\mathrm{sub}}} \sum_{\ell=1}^{n_{\mathrm{sub}}} \left(\frac{F_\ell}{m|g|}\right)^2, w_{\mathrm{ctrl}}\|a_t\|_2^2\right]. \tag{24}$$

The costs measure native velocity-limit violations, squared foot-floor contact load normalized by body weight and averaged over physics substeps, and control effort. Here $v_{\max} = 0.7402\,\mathrm{m\,s^{-1}}$, $w_{\mathrm{load}} = 0.01$, and $w_{\mathrm{ctrl}} = 0.1$.

MO-LunarLander. For the upstream reward vector $r_t = (r_{0,t}, \ldots, r_{3,t})$, $r_{0,t}$ is the terminal landing outcome, $r_{1,t}$ is the shaping reward, and $r_{2,t}$ and $r_{3,t}$ are the non-positive main- and side-engine terms. We use

$$R_t = r_{0,t} + r_{1,t}, \qquad c_{\mathrm{main},t} = \max(0, -r_{2,t}), \qquad c_{\mathrm{side},t} = \max(0, -r_{3,t}). \tag{25}$$

EV2Gym. The reward is energy-trading profit without a satisfaction penalty. If $\mathcal{D}_t$ is the set of EVs departing at step t, N_{EV} is the number of EVs in the episode, and $q_j \in [0, 1]$ is the clipped ratio of current capacity to the configured target capacity at departure, the step cost is

$$c_t = \frac{1}{N_{\mathrm{EV}}} \sum_{j \in \mathcal{D}_t} (1 - q_j). \tag{26}$$

The target state of charge (SOC) is 0.8. Normalization by the number of EVs retained in the episode makes the undiscounted episode sum the average departure shortfall. We use ten homogeneous charging stations, one action per station, a 160-step horizon with 15 minutes per step, and a 600 kW transformer limit. The 100-dimensional observation contains normalized time and aggregate power, a 48-step price lookahead, and, for each charger, a connection indicator, state of charge, target-capacity gap, time to announced departure, and charging slack.

Only EV profiles that can reach the target by their announced departure are retained. For early-departure cap $\Delta_{\max}$, the actual departure shift is sampled uniformly over the nonnegative integer shifts bounded by $\Delta_{\max}$ and by the EV's charging slack. The policy observes the announced departure but not the sampled shift. Policies are trained with an unobserved early-departure cap of four steps and evaluated with caps $\{0, 4, 8\}$ using matched episode seeds.

Task	Ordered costs	Cost limits	Reward scale s_R	Horizon	Training steps
PointCircle1	boundary, drive, turn	$[25, 133, 53]$	1.69	500	10M
CarGoal2	hazards, vase contact, vase velocity	$[6, 6, 19]$	0.11	1000	10M
HalfCheetahVelocity	velocity, pitch, control	$[25, 44, 127]$	217.72	1000	10M
HopperVelocity	velocity, landing load, control	$[25, 16, 41]$	62.92	1000	10M
MO-LunarLander	main engine, side engine	$[92, 27]$	30.37	1000	5M
EV2Gym	departure shortfall	0.1	48.84	160	5M

Table 5: Task settings. Limits apply to discounted episodic costs, and s_R is the fixed reward scale calibrated in Appendix A.2.

A.2 Cost-Limit and Reward-Scale Calibration

PointCircle1, HalfCheetahVelocity, and HopperVelocity retain the native binary-cost limit of 25. The auxiliary continuous-cost limits of PointCircle1, HalfCheetahVelocity, and HopperVelocity are one half of the median reward-only PPO per-policy mean discounted cost, rounded to the nearest integer. The MO-LunarLander limits use the empirical threshold jointly satisfied by 25% of successful reward-only PPO trajectories. EV2Gym uses the configured discounted-cost limit 0.1; we do not identify this discounted limit with an undiscounted satisfaction percentage.

For CarGoal2, four reward-only PPO final policies are evaluated deterministically for $1{,}000$ episodes each, with horizon $1{,}000$ and discount 0.999. Let M_0 and M_j denote the medians across policies of the per-policy mean discounted native OR cost and component-j cost, respectively. With native OR-cost limit $d_0 = 25$, we set $d_j = \mathrm{round}(d_0 M_j / M_0)$.

The fixed reward scale pools within-policy return variation from $K = 4$ reward-only PPO final policies trained with different seeds, each evaluated deterministically for $1{,}000$ episodes with discount 0.999:

$$s_R = \sqrt{\frac{\sum_{k=1}^{K} (N_k - 1)\hat{\sigma}_k^2}{\sum_{k=1}^{K} (N_k - 1)}}. \tag{27}$$

Here, $\hat{\sigma}_k$ is the sample SD of discounted returns from policy k and N_k is its episode count. Scales are rounded to two decimal places and fixed throughout constrained training.

A.3 Algorithm and Evaluation Settings

All learned methods use the component order and limits in Table 5, a reward and cost discount of 0.999, and are evaluated at the final checkpoint of each training run. In multi-cost tasks, each baseline maintains a separate cost critic and constraint for every component. The marginal risk-sensitive baselines apply their risk parameters componentwise.

Setting	SAC-Lag	PPO-Lag	SAC-PID	SDAC	SRCPO	WCSAC	JRCPO
Hidden sizes	$[256, 256]$	$[64, 64]$	$[256, 256]$	$[512, 512]$	$[512, 512]$	$[512, 512]$	$[512, 512]$
Activation	ReLU	Tanh	ReLU	LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU
Actor LR/update	3×10^{-4}	3×10^{-4}	3×10^{-4}	trust region	trust region	3×10^{-4}	3×10^{-4}
Critic LR	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Batch size	256	64	256	–	–	256	256
PPO clip	–	0.2	–	–	–	–	–
KL limit	–	0.02	–	10^{-3}	10^{-3}	–	–
Entropy setup	fixed 0.2	fixed 0	fixed 0.2	–	–	adaptive, LR 10^{-3}	fixed 0.05
Lagrange LR	10^{-5}	3.5×10^{-2}	–	–	–	10^{-3}	10^{-2} (MO-LunarLander, EV2Gym: 10^{-3})
PID gains (K_P, K_I, K_D)	–	–	$(10^{-6}, 10^{-7}, 10^{-7})$	–	–	–	–
Multiplier warm-up	200,000 steps	–	200,000 steps	–	–	–	100 epochs
confidence α	–	–	–	0.1	0.1	0.1	0.1
Distributional critics	–	–	–	2×25 quantiles	2×25 quantiles	2×25 quantiles	2 Gaussians
Target quantiles/samples	–	–	–	50 quantiles	50 quantiles	–	–
TD(λ) coefficient	–	–	–	0.97	0.97	–	–
Sampler LR	–	–	–	–	3×10^{-4}	–	–

Table 6: Implementation settings. Hidden sizes apply to actor and critic networks; – means not applicable.

AFAP charges each connected EV at the maximum rate up to the target SOC of 0.8, reducing the final charging action to the remaining target gap and assigning zero action to disconnected chargers. It uses no training and is evaluated on the same episode seeds and noise levels as the learned policies.

For N evaluation episodes, cost CVaR averages the largest $\lceil 0.1N \rceil$ discounted component costs. Mean and CVaR are computed per trained policy and averaged across training seeds; reported SDs are sample SDs of the per-policy means.

B Additional Results

Table 7: Dispersion across five multi-cost tasks, extending Table 4.

Task	SAC-Lag	PPO-Lag	SAC-PID	SDAC	WCSAC	**JRCPO**
PointCircle1	1.715 ±0.879	2.827 ±2.577	2.305 ±1.168	2.732 ±1.047	1.795 ±1.230	0.150 ±0.013
CarGoal2	1.147 ±0.034	2.432 ±0.923	1.202 ±0.053	3.178 ±1.809	11.545 ±11.178	2.124 ±0.691
HalfCheetahVelocity	1.504 ±2.863	0.183 ±0.024	1.199 ±1.618	0.175 ±0.200	0.453 ±0.757	0.054 ±0.037
HopperVelocity	1.559 ±1.684	1.109 ±0.876	4.179 ±2.297	0.237 ±0.324	0.020 ±0.013	0.497 ±0.495
MO-LunarLander	0.735 ±0.661	3.995 ±0.776	1.700 ±0.937	3.721 ±1.276	1.241 ±0.491	0.743 ±0.371

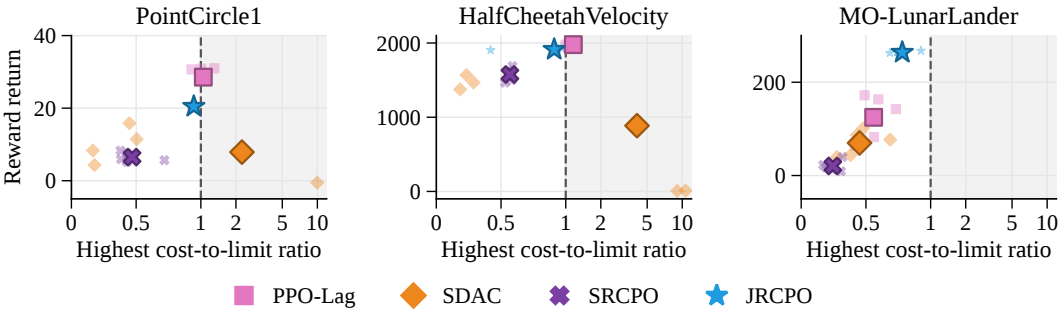

Figure 4: Reward-cost return trade-offs in three multi-cost tasks. Large and faint markers show means over five training seeds and individual seeds, respectively. For each method and task, all points use the cost component with the highest across-seed mean cost-to-limit ratio. The x-axis uses symlog scaling with linear threshold 1.

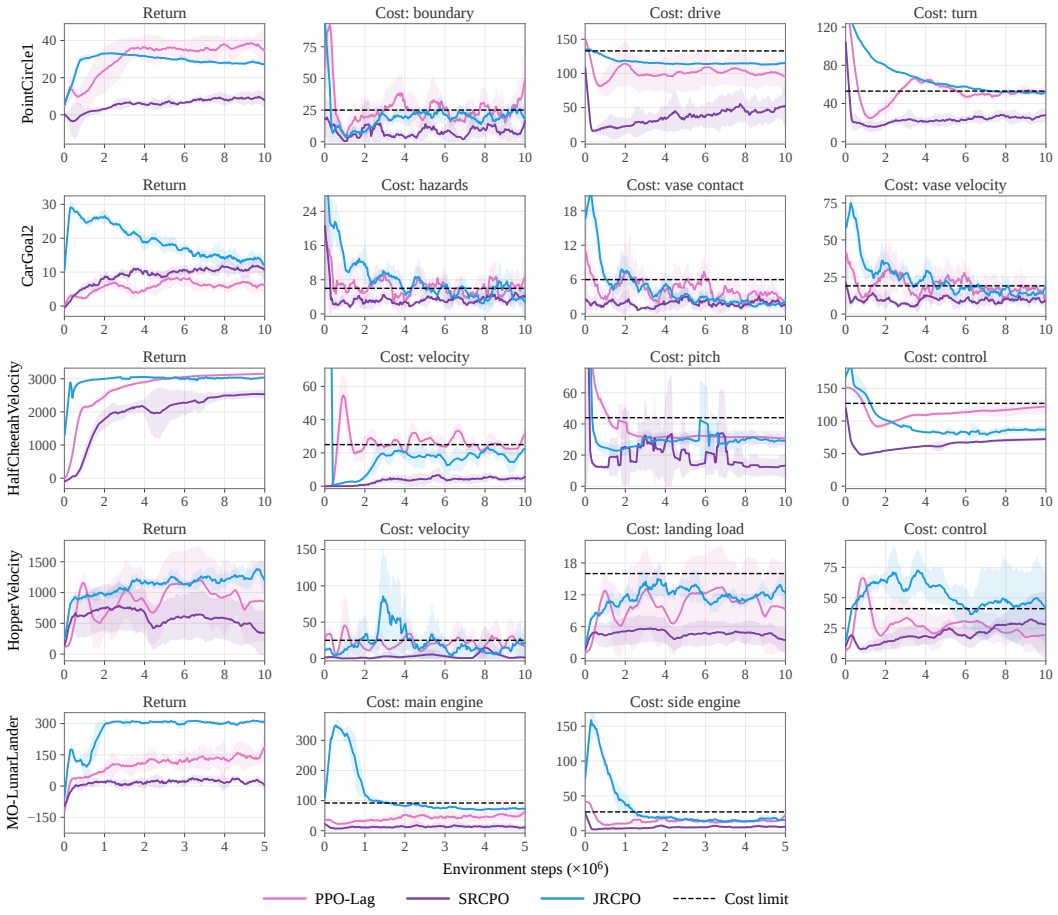

Figure 5: Learning curves of PPO-Lag, SRCPO, and JRCPO on five multi-cost tasks. Each row shows one task, with the undiscounted reward return in the first column and the discounted cost return of each component in the remaining columns. Lines denote the mean of the smoothed curves over five seeds, and shaded regions denote one standard deviation.