A Properties of coherent distortion risk measures

Majumdar and Pavone [30] proposed a set of six axioms to characterize desirable properties of risk measures in the context of robotics.

A1. Monotonicity: If $Z, Z' \in \mathcal{Z}$ and $Z \leq Z'$ almost everywhere, then $\rho(Z) \leq \rho(Z')$.

A2. Translation invariance: If $\alpha \in \mathbb{R}$ and $Z \in \mathcal{Z}$, then $\rho(Z + \alpha) = \rho(Z) + \alpha$.

A3. Positive homogeneity: If $\tau \geq 0$ and $Z \in \mathcal{Z}$, then $\rho(\tau Z) = \tau \rho(Z)$.

A4. Convexity: If $\lambda \in [0, 1]$ and $Z, Z' \in \mathcal{Z}$, then $\rho(\lambda Z + (1 - \lambda)Z') \leq \lambda \rho(Z) + (1 - \lambda)\rho(Z')$.

A5. Comonotonic additivity: If $Z, Z' \in \mathcal{Z}$ are comonotonic, then $\rho(Z + Z') = \rho(Z) + \rho(Z')$.

A6. Law invariance: If $Z, Z' \in \mathcal{Z}$ are identically distributed, then $\rho(Z) = \rho(Z')$.

See Majumdar and Pavone [30] for a discussion on the intuition behind these axioms. Note that coherent risk measures [6] satisfy Axioms A1–A4, distortion risk measures [17, 50] satisfy Axioms A1–A3 and Axioms A5–A6, and coherent distortion risk measures satisfy all six axioms.

The properties of coherent risk measures also lead to a useful dual representation.

Lemma 2 (Shapiro et al. [42])**.** *Let ρ be a proper, real-valued coherent risk measure. Then, for any $Z \in \mathcal{Z}$ we have that*

$$\rho(Z) = \sup_{\beta_{s,a} \in \mathcal{U}_{s,a}} \mathbb{E}_{\beta_{s,a}}[Z],$$

where $\mathbb{E}_{\beta_{s,a}}[\cdot]$ represents expectation with respect to the probability measure $\beta_{s,a} \in P(\mathcal{M})$, and

$$\mathcal{U}_{s,a} \subseteq \{\beta_{s,a} \in P(\mathcal{M}) \mid \beta_{s,a} = \xi_{s,a}\mu_{s,a},\ \xi_{s,a} \in \mathcal{Z}^*\}$$

is a convex, bounded, and weakly closed set that depends on ρ.*

See Shapiro et al. [42] for a general treatment of this result.

B Proofs

In this section, we prove all results related to the RAMU cost Bellman operator $\mathcal{T}^\pi_{\rho,c}$. Using the fact that $\rho^+(Z) = -\rho(-Z)$ for a coherent distortion risk measure ρ on a cost random variable, all results related to the RAMU reward Bellman operator follow by an appropriate change in sign.

B.1 Proof of Lemma 1

Proof. Starting from the definition of $\mathcal{T}^\pi_{\rho,c}$ in Definition 3, we have that

$$\begin{aligned}
\mathcal{T}^\pi_{\rho,c}Q(s,a) &= c(s,a) + \gamma \underset{p_{s,a}\sim\mu_{s,a}}{\rho} \left(\mathop{\mathbb{E}}_{s'\sim p_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} [Q(s',a')] \right] \right) \\
&= c(s,a) + \underset{p_{s,a}\sim\mu_{s,a}}{\rho} \left(\gamma \mathop{\mathbb{E}}_{s'\sim p_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} [Q(s',a')] \right] \right) \\
&= \underset{p_{s,a}\sim\mu_{s,a}}{\rho} \left(c(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim p_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} [Q(s',a')] \right] \right) \\
&= \underset{p_{s,a}\sim\mu_{s,a}}{\rho} \left(\mathcal{T}^\pi_{p,c}Q(s,a) \right),
\end{aligned}$$

which proves the result. Note that the second equality follows from the positive homogeneity of ρ (Axiom A3), the third equality follows from the translation invariance of ρ (Axiom A2), and the fourth equality follows from the definition of the standard cost Bellman operator $\mathcal{T}^\pi_{p,c}$. □

B.2 Proof of Theorem 1

Proof. First, we show that $\mathcal{T}^\pi_{\rho,c}$ is equivalent to a distributionally robust Bellman operator. For a given state-action pair, we apply Lemma 2 to the risk measure that appears in the formulation of $\mathcal{T}^\pi_{\rho,c}$

given by Lemma 1. By doing so, we have that

$$\begin{aligned}\mathcal{T}^{\pi}_{\rho,c}Q(s,a) &= \underset{p_{s,a}\sim\mu_{s,a}}{\rho}\left(\mathcal{T}^{\pi}_{p,c}Q(s,a)\right)\\ &= \sup_{\beta_{s,a}\in\mathcal{U}_{s,a}} \underset{p_{s,a}\sim\beta_{s,a}}{\mathbb{E}}\left[\mathcal{T}^{\pi}_{p,c}Q(s,a)\right]\\ &= c(s,a)+\gamma \sup_{\beta_{s,a}\in\mathcal{U}_{s,a}} \underset{p_{s,a}\sim\beta_{s,a}}{\mathbb{E}}\left[\underset{s'\sim p_{s,a}}{\mathbb{E}}\left[\underset{a'\sim\pi(\cdot|s')}{\mathbb{E}}\left[Q(s',a')\right]\right]\right],\end{aligned}$$

where $\mathcal{U}_{s,a}$ is defined in Lemma 2. Therefore, $\mathcal{T}^{\pi}_{\rho,c}$ has the same form as a distributionally robust Bellman operator [53, 56] with the ambiguity set $\mathcal{U}=\bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathcal{U}_{s,a}$. The RAMU cost Q function $Q^{\pi}_{\rho,c}(s,a)$ is the fixed point of $\mathcal{T}^{\pi}_{\rho,c}$, so it is equivalent to a distributionally robust Q function with ambiguity set $\mathcal{U}$. Using the rectangularity of $\mathcal{U}$, we can write this succinctly as

$$Q^{\pi}_{\rho,c}(s,a)=\sup_{\beta\in\mathcal{U}}\underset{p\sim\beta}{\mathbb{E}}\left[Q^{\pi}_{p,c}(s,a)\right].$$

Then, using the definition of $J_{\rho,c}(\pi)$ we have that

$$\begin{aligned}J_{\rho,c}(\pi) &= \underset{s\sim d_0}{\mathbb{E}}\left[\underset{a\sim\pi(\cdot|s)}{\mathbb{E}}\left[Q^{\pi}_{\rho,c}(s,a)\right]\right]\\ &= \underset{s\sim d_0}{\mathbb{E}}\left[\underset{a\sim\pi(\cdot|s)}{\mathbb{E}}\left[\sup_{\beta\in\mathcal{U}}\underset{p\sim\beta}{\mathbb{E}}\left[Q^{\pi}_{p,c}(s,a)\right]\right]\right]\\ &= \sup_{\beta\in\mathcal{U}}\underset{p\sim\beta}{\mathbb{E}}\left[\underset{s\sim d_0}{\mathbb{E}}\left[\underset{a\sim\pi(\cdot|s)}{\mathbb{E}}\left[Q^{\pi}_{p,c}(s,a)\right]\right]\right]\\ &= \sup_{\beta\in\mathcal{U}}\underset{p\sim\beta}{\mathbb{E}}\left[J_{p,c}(\pi)\right],\end{aligned}$$

where we can move the optimization over $\mathcal{U}$ outside of the expectation operators due to rectangularity.

We can use similar techniques to show that $\mathcal{T}^{\pi}_{\rho^+,r}$ has the same form as a distributionally robust Bellman operator with the ambiguity set $\mathcal{U}^+=\bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathcal{U}^+_{s,a}$, and

$$J_{\rho^+,r}(\pi)=\inf_{\beta\in\mathcal{U}^+}\underset{p\sim\beta}{\mathbb{E}}\left[J_{p,r}(\pi)\right].$$

Therefore, we have that the RAMU safe RL problem in (3) is equivalent to (6). □

B.3 Proof of Corollary 1

Given the equivalence of $\mathcal{T}^{\pi}_{\rho^+,r}$ and $\mathcal{T}^{\pi}_{\rho,c}$ to distributionally robust Bellman operators as shown in Theorem 1, Corollary 1 follows from results in Xu and Mannor [53] and Yu and Xu [56]. We include a proof for completeness.

Proof. Due to the linearity of the expectation operator, for a given $\beta_{s,a}\in\mathcal{U}_{s,a}$ we have that

$$\underset{p_{s,a}\sim\beta_{s,a}}{\mathbb{E}}\left[\underset{s'\sim p_{s,a}}{\mathbb{E}}\left[\underset{a'\sim\pi(\cdot|s')}{\mathbb{E}}\left[Q(s',a')\right]\right]\right]=\underset{s'\sim\bar{p}^{\beta}_{s,a}}{\mathbb{E}}\left[\underset{a'\sim\pi(\cdot|s')}{\mathbb{E}}\left[Q(s',a')\right]\right],$$

where $\bar{p}^{\beta}_{s,a}=\mathbb{E}_{p_{s,a}\sim\beta_{s,a}}[p_{s,a}]\in P(\mathcal{S})$ represents a mixture transition model determined by $\beta_{s,a}$. Therefore, starting from the result in Theorem 1, we can write

$$\begin{aligned}\mathcal{T}^{\pi}_{\rho,c}Q(s,a) &= c(s,a)+\gamma\sup_{\beta_{s,a}\in\mathcal{U}_{s,a}}\underset{p_{s,a}\sim\beta_{s,a}}{\mathbb{E}}\left[\underset{s'\sim p_{s,a}}{\mathbb{E}}\left[\underset{a'\sim\pi(\cdot|s')}{\mathbb{E}}\left[Q(s',a')\right]\right]\right]\\ &= c(s,a)+\gamma\sup_{\bar{p}^{\beta}_{s,a}\in\mathcal{P}_{s,a}}\underset{s'\sim\bar{p}^{\beta}_{s,a}}{\mathbb{E}}\left[\underset{a'\sim\pi(\cdot|s')}{\mathbb{E}}\left[Q(s',a')\right]\right],\end{aligned}$$

where

$$\mathcal{P}_{s,a}=\left\{\bar{p}^{\beta}_{s,a}\in P(\mathcal{S})\mid\bar{p}^{\beta}_{s,a}=\underset{p_{s,a}\sim\beta_{s,a}}{\mathbb{E}}\left[p_{s,a}\right],\ \beta_{s,a}\in\mathcal{U}_{s,a}\right\}.$$

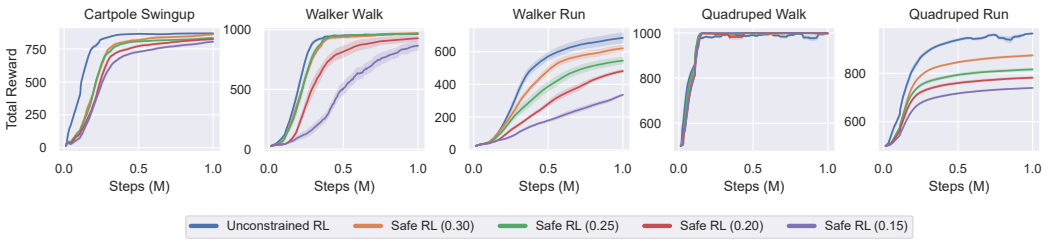

Figure 3: Hyperparameter sweep of safety coefficient. Value in parentheses represents safety coefficient used for training in safe RL. Shading denotes half of one standard error across policies.

As a result, $\mathcal{T}^{\pi}_{\rho,c}$ has the same form as a robust Bellman operator [21, 34] with the uncertainty set $\mathcal{P} = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}_{s,a}$.

Consider Q functions $Q^{(1)}$ and $Q^{(2)}$, and denote the sup-norm by

$$\|Q^{(1)} - Q^{(2)}\|_\infty = \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left| Q^{(1)}(s,a) - Q^{(2)}(s,a) \right|.$$

Fix $\epsilon > 0$ and consider $(s,a) \in \mathcal{S}\times\mathcal{A}$. Then, there exists $\bar{p}^{(1)}_{s,a} \in \mathcal{P}_{s,a}$ such that

$$\mathop{\mathbb{E}}_{s'\sim\bar{p}^{(1)}_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} \left[Q^{(1)}(s',a') \right] \right] \geq \sup_{\bar{p}^{\beta}_{s,a}\in\mathcal{P}_{s,a}} \mathop{\mathbb{E}}_{s'\sim\bar{p}^{\beta}_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} \left[Q^{(1)}(s',a') \right] \right] - \epsilon.$$

We have that

$$\begin{aligned}
&\mathcal{T}^{\pi}_{\rho,c} Q^{(1)}(s,a) - \mathcal{T}^{\pi}_{\rho,c} Q^{(2)}(s,a) \\
&= \gamma \left(\sup_{\bar{p}^{\beta}_{s,a}\in\mathcal{P}_{s,a}} \mathop{\mathbb{E}}_{s'\sim\bar{p}^{\beta}_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} \left[Q^{(1)}(s',a') \right] \right] - \sup_{\bar{p}^{\beta}_{s,a}\in\mathcal{P}_{s,a}} \mathop{\mathbb{E}}_{s'\sim\bar{p}^{\beta}_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} \left[Q^{(2)}(s',a') \right] \right] \right) \\
&\leq \gamma \left(\mathop{\mathbb{E}}_{s'\sim\bar{p}^{(1)}_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} \left[Q^{(1)}(s',a') \right] \right] + \epsilon - \mathop{\mathbb{E}}_{s'\sim\bar{p}^{(1)}_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} \left[Q^{(2)}(s',a') \right] \right] \right) \\
&= \gamma \mathop{\mathbb{E}}_{s'\sim\bar{p}^{(1)}_{s,a}} \left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')} \left[Q^{(1)}(s',a') - Q^{(2)}(s',a') \right] \right] + \gamma\epsilon \\
&\leq \gamma \|Q^{(1)} - Q^{(2)}\|_\infty + \gamma\epsilon.
\end{aligned}$$

A similar argument can be used to show that

$$-\gamma\|Q^{(1)} - Q^{(2)}\|_\infty - \gamma\epsilon \leq \mathcal{T}^{\pi}_{\rho,c} Q^{(1)}(s,a) - \mathcal{T}^{\pi}_{\rho,c} Q^{(2)}(s,a),$$

so we have that

$$\left| \mathcal{T}^{\pi}_{\rho,c} Q^{(1)}(s,a) - \mathcal{T}^{\pi}_{\rho,c} Q^{(2)}(s,a) \right| \leq \gamma\|Q^{(1)} - Q^{(2)}\|_\infty + \gamma\epsilon.$$

By applying a supremum over state-action pairs on the left-hand side, we obtain

$$\|\mathcal{T}^{\pi}_{\rho,c} Q^{(1)} - \mathcal{T}^{\pi}_{\rho,c} Q^{(2)}\|_\infty \leq \gamma\|Q^{(1)} - Q^{(2)}\|_\infty + \gamma\epsilon.$$

Finally, since $\epsilon > 0$ was arbitrary, we have shown that $\mathcal{T}^{\pi}_{\rho,c}$ is a γ-contraction in the sup-norm. □

C Implementation details and additional experimental results

Safety constraints and environment perturbations In all of our experiments, we consider the problem of optimizing a task objective while satisfying a safety constraint. We focus on a single safety constraint corresponding to a cost function defined in the Real-World RL Suite for each task,

Table 2: Safety constraints for all tasks

Task	Safety Constraint	Safety Coefficient
Cartpole Swingup	Slider Position	0.30
Walker Walk	Joint Velocity	0.25
Walker Run	Joint Velocity	0.30
Quadruped Walk	Joint Angle	0.15
Quadruped Run	Joint Angle	0.30

Table 3: Perturbation ranges for test environments across domains

Domain	Perturbation Parameter	Nominal Value	Test Range
Cartpole	Pole Length	1.00	$[0.75, 1.25]$
Walker	Torso Length	0.30	$[0.10, 0.50]$
Quadruped	Torso Density	1,000	$[500, 1{,}500]$

and we consider a safety budget of $B = 100$. The safety constraints used for each task are described in Table 2. In the Cartpole domain, costs are applied when the slider is outside of a specified range. In the Walker domain, costs are applied for large joint velocities. In the Quadruped domain, costs are applied for large joint angles. See Dulac-Arnold et al. [19] for detailed definitions of each safety constraint.

The definitions of these cost functions depend on a safety coefficient in $[0, 1]$, which determines the range of outcomes that lead to constraint violations and therefore controls how difficult it will be to satisfy safety constraints corresponding to these cost functions. As the safety coefficient decreases, the range of safe outcomes also decreases and the safety constraint becomes more difficult to satisfy. In order to consider safe RL tasks with difficult safety constraints where strong performance is still possible, we selected the value of this safety constraint in the range of $[0.15, 0.20, 0.25, 0.30]$ for each task based on the performance of the baseline safe RL algorithm CRPO compared to the unconstrained algorithm MPO. Figure 3 shows total rewards throughout training for each task across this range of safety coefficients. We selected the most difficult cost definition in this range (i.e., lowest safety coefficient value) where CRPO is still able to achieve the same total rewards as MPO (or the value that leads to the smallest gap between the two in the case of Walker Run and Quadruped Run). The resulting safety coefficients used for our experiments are listed in Table 2.

In order to evaluate the robustness of our learned policies, we generate a range of test environments for each task based on perturbing a simulator parameter in the Real-World RL Suite. See Table 3 for the perturbation parameters and corresponding ranges considered in our experiments. The test range for each domain is centered around the nominal parameter value that defines the single training environment used for all experiments except domain randomization. See Figure 4 for detailed results of the risk-averse and risk-neutral versions of our RAMU framework across all tasks and environment perturbations.

Domain randomization Domain randomization requires a training distribution over a range of environments, which is typically defined by considering a range of simulator parameters. For the in-distribution version of domain randomization considered in our experiments, we apply a uniform distribution over a subset of the test environments defined in Table 3. In particular, we consider the middle 50% of test environment parameter values centered around the nominal environment value for training. In the out-of-distribution version of domain randomization, on the other hand, we consider a different perturbation parameter from the one varied at test time. We apply a uniform distribution over a range of values for this alternate parameter centered around the value in the nominal environment. Therefore, the only environment shared between the set of test environments and the set of training environments used for out-of-distribution domain randomization is the nominal environment. See Table 4 for details on the parameters and corresponding ranges used for training in domain randomization.

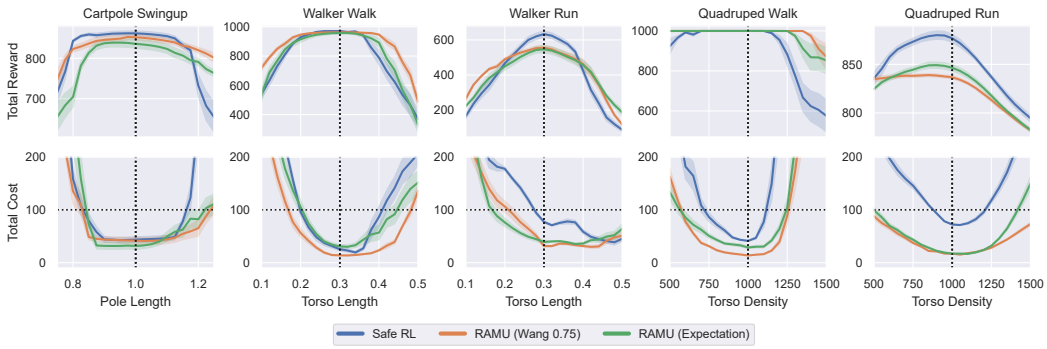

Figure 4: Comparison with standard safe RL across tasks and test environments. RAMU algorithms use the risk measure in parentheses applied to both the objective and constraint. Shading denotes half of one standard error across policies. Vertical dotted lines represent nominal training environment. Top: Total reward. Bottom: Total cost, where horizontal dotted lines represent safety budget.

Table 4: Perturbation parameters and ranges for domain randomization across domains

Domain	Perturbation Parameter	Nominal Value	Training Range
In-Distribution			
Cartpole	Pole Length	1.00	$[0.875, 1.125]$
Walker	Torso Length	0.30	$[0.20, 0.40]$
Quadruped	Torso Density	1,000	$[750, 1{,}250]$
Out-of-Distribution			
Cartpole	Pole Mass	0.10	$[0.05, 0.15]$
Walker	Contact Friction	0.70	$[0.40, 1.00]$
Quadruped	Contact Friction	1.50	$[1.00, 2.00]$

We include the results for domain randomization across all tasks and environment perturbations in Figure 5. Across all tasks, we observe that our RAMU framework leads to similar or improved constraint satisfaction compared to in-distribution domain randomization, while only using one training environment. In addition, our framework consistently outperforms out-of-distribution domain randomization, which provides little benefit compared to standard safe RL due to its misspecified training distribution.

Adversarial reinforcement learning In order to implement the action-robust PR-MDP framework, we must train an adversarial policy. We represent the adversarial policy using the same structure and neural network architecture as our main policy, and we train the adversarial policy to maximize total costs using MPO. Using the default setting in Tessler et al. [46], we apply one adversary update for every 10 policy updates.

We include the results for adversarial RL across all tasks and environment perturbations in Figure 6, where adversarial RL is evaluated without adversarial interventions. We see that adversarial RL leads to robust safety in some cases, such as the two Quadruped tasks. However, in other tasks such as Cartpole Swingup, safety constraint satisfaction is not as robust. Safety also comes at the cost of conservative performance in some tasks, as evidenced by the total rewards achieved by adversarial RL in Walker Run and Quadruped Run. Overall, our RAMU framework achieves similar performance to adversarial RL, without the drawbacks associated with adversarial methods that preclude their use in some real-world settings.

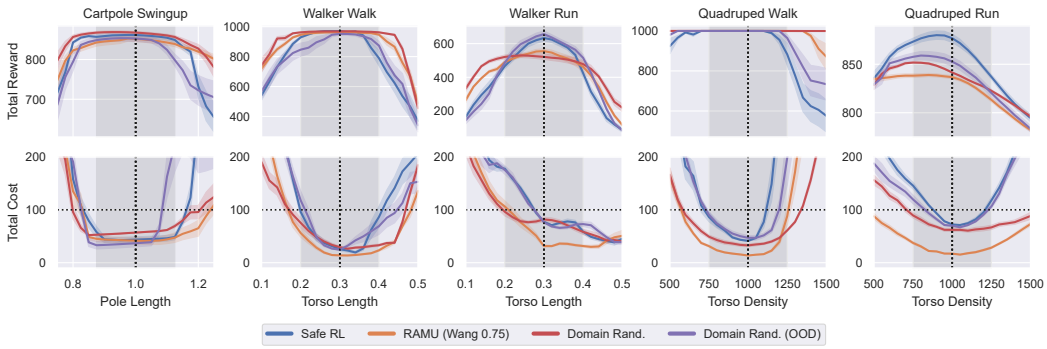

Figure 5: Comparison with domain randomization across tasks and test environments. Grey shaded area denotes the training range for in-distribution domain randomization. Shading denotes half of one standard error across policies. Vertical dotted lines represent nominal training environment. Top: Total reward. Bottom: Total cost, where horizontal dotted lines represent safety budget.

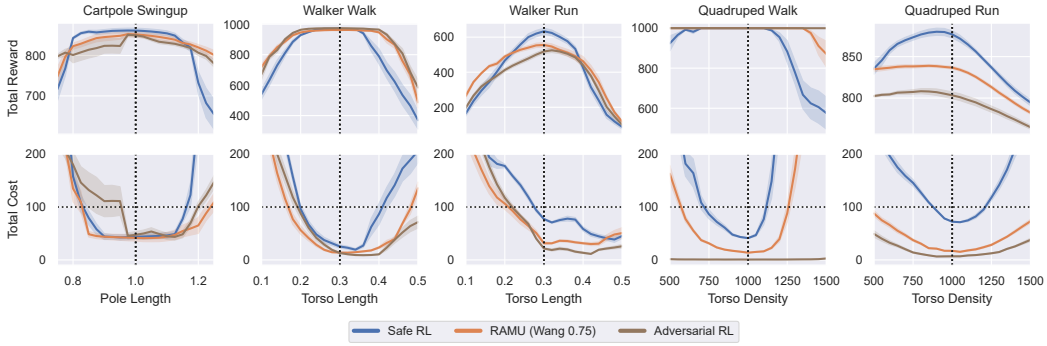

Figure 6: Comparison with adversarial RL across tasks and test environments. Performance of adversarial RL is evaluated without adversarial interventions. Shading denotes half of one standard error across policies. Vertical dotted lines represent nominal training environment. Top: Total reward. Bottom: Total cost, where horizontal dotted lines represent safety budget.

Network architectures and algorithm hyperparameters In our experiments, we consider neural network representations of the policy and critics. Each of these neural networks contains 3 hidden layers of 256 units with ELU activations. In addition, we apply layer normalization followed by a tanh activation after the first layer of these networks as proposed in Abdolmaleki et al. [2]. We consider a multivariate Gaussian policy, where at a given state we have $\pi(a \mid s) = \mathcal{N}(\mu(s), \Sigma(s))$ where $\mu(s)$ and $\Sigma(s)$ represent outputs of the policy network. $\Sigma(s)$ is a diagonal covariance matrix, whose diagonal elements are calculated by applying the softplus operator to the outputs of the neural network. We parameterize the reward and cost critics with separate neural networks. In addition, we consider target networks that are updated as an exponential moving average with parameter $\tau = 5\text{e-}3$.

We consider CRPO [55] as the baseline safe RL algorithm in all of our experiments, which immediately switches between maximizing rewards and minimizing costs at every update based on the value of the safety constraint. If the sample-average estimate of the safety constraint for the current batch of data satisfies the safety budget, we update the policy to maximize rewards. Otherwise, we update the policy to minimize costs.

After CRPO determines the appropriate objective for the current batch of data, we apply MPO [1] to calculate policy updates. MPO calculates a non-parametric policy update based on the KL divergence parameter ϵ_{KL}, and then takes a step towards this non-parametric policy while constraining the KL divergence from updating the mean by β_μ and the KL divergence from updating the covariance matrix

Table 5: Network architectures and algorithm hyperparameters used in experiments

General	
Batch size per update	256
Updates per environment step	1
Discount rate (γ)	0.99
Target network exponential moving average (τ)	5e-3
Policy	
Layer sizes	256, 256, 256
Layer activations	ELU
Layer norm + tanh on first layer	Yes
Initial standard deviation	0.3
Learning rate	1e-4
Non-parametric KL (ϵ_{KL})	0.10
Action penalty KL	1e-3
Action samples per update	20
Parametric mean KL (β_μ)	0.01
Parametric covariance KL (β_Σ)	1e-5
Parametric KL dual learning rate	0.01
Critics	
Layer sizes	256, 256, 256
Layer activations	ELU
Layer norm + tanh on first layer	Yes
Learning rate	1e-4
RAMU	
Transition model samples per data point (n)	5
Latent variable hyperparameter (ϵ)	0.10

by β_Σ. We consider per-dimension KL divergence constraints by dividing these parameter values by the number of action dimensions, and we penalize actions outside of the feasible action limits using the multi-objective MPO framework [2] as suggested in Hoffman et al. [20]. In order to avoid potential issues related to the immediate switching between reward and cost objectives throughout training, we completely solve for the temperature parameter of the non-parametric target policy in MPO at every update as done in Liu et al. [28]. See Table 5 for the default hyperparameter values used in our experiments, which are based on default values considered in Hoffman et al. [20].

For our RAMU framework, the latent variable hyperparameter ϵ controls the definition of the distribution $\mu_{s,a}$ over transition models. Figure 7 shows the performance of our RAMU framework in Walker Run and Quadruped Run for $\epsilon \in [0.05, 0.10, 0.15, 0.20]$. A larger value of ϵ leads to a distribution over a wider range of transition models, which results in a more robust approach when combined with a risk-averse perspective on model uncertainty. We see in Figure 7 that our algorithm more robustly satisfies safety constraints as ϵ increases, but this robustness also leads to a decrease in total rewards. We consider $\epsilon = 0.10$ in our experiments, as it achieves strong constraint satisfaction without a meaningful decrease in rewards. Finally, for computational efficiency we consider $n = 5$ samples of transition models per data point to calculate sample-based Bellman targets in our RAMU framework, as we did not observe meaningful improvements in performance from considering a larger number of samples.

Computational resources All experiments were run on a Linux cluster with 2.9 GHz Intel Gold processors and NVIDIA A40 and A100 GPUs. The Real-World RL Suite is available under the Apache License 2.0. We trained policies for 1 million steps across 5 random seeds, which required

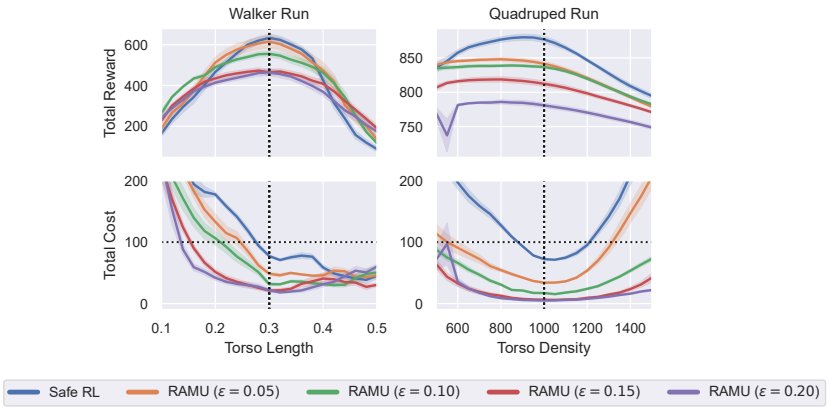

Figure 7: Hyperparameter sweep of latent variable hyperparameter ϵ on Walker Run and Quadruped Run. RAMU algorithms use the Wang transform with $\eta = 0.75$ applied to both the objective and constraint. Shading denotes half of one standard error across policies. Vertical dotted lines represent nominal training environment. Top: Total reward. Bottom: Total cost, where horizontal dotted lines represent safety budget.

approximately one day of wall-clock time on a single GPU for each combination of algorithm and task using code that has not been optimized for execution speed.