Distributionally Robust Self Paced Curriculum Reinforcement Learning

Anirudh Satheesh
University of Maryland
College Park, MD 20742
anirudhs@terpmail.umd.edu

Keenan Powell
University of Maryland
College Park, MD 20742
kpowell1@terpmail.umd.edu

Vaneet Aggarwal
Purdue University
West Lafayette, IN 47907
vaneet@purdue.edu

Abstract

A central challenge in reinforcement learning is that policies trained in controlled environments often fail under distribution shifts at deployment into real-world environments. Distributionally Robust Reinforcement Learning (DRRL) addresses this by optimizing for worst-case performance within an uncertainty set defined by a robustness budget ϵ. However, fixing ϵ results in a tradeoff between performance and robustness: small values yield high nominal performance but weak robustness, while large values can result in instability and overly conservative policies. We propose Distributionally Robust Self-Paced Curriculum Reinforcement Learning (DR-SPCRL), a method that overcomes this limitation by treating ϵ as a continuous curriculum. DR-SPCRL adaptively schedules the robustness budget according to the agent's progress, enabling a balance between nominal and robust performance. Empirical results across multiple environments demonstrate that DR-SPCRL not only stabilizes training but also achieves a superior robustness–performance trade-off, yielding an average 11.8% increase in episodic return under varying perturbations compared to fixed or heuristic scheduling strategies, and achieving approximately $1.9\times$ the performance of the corresponding nominal RL algorithms.

1 Introduction

Curriculum Reinforcement Learning (CRL) has emerged as a powerful paradigm for accelerating and stabilizing the training of reinforcement learning (RL) agents. By structuring training from simpler to progressively harder tasks, CRL allows agents to acquire complex skills more reliably and to generalize better to unseen situations [1, 2]. Instead of exposing an agent to the full complexity of a target task from the beginning, CRL gradually introduces a series of intermediate tasks tailored to the agent's current level of proficiency.

While CRL improves learning efficiency and stability, it primarily focuses on generating and learning semantic or goal-specific tasks and, as a result, cannot provide guarantees on performance across arbitrary environment parameterizations. In many real-world applications, policies must contend with unmodeled dynamics, sensor noise, and physical variations between training and deployment, known as the sim-to-real problem. These sources of uncertainty can cause dramatic performance degradation even for policies that performed well during training. To address this, Distributionally Robust Reinforcement Learning (DRRL) [3] provides a principled framework for learning policies that maximize worst-case returns over a prescribed uncertainty set. The resulting policies are explicitly designed to be resilient to model mismatch, in contrast to standard RL approaches that optimize performance only in the nominal environment.

In DRRL, the size of the uncertainty set is determined by a robustness budget ϵ, which specifies the allowable deviation between the nominal model and its perturbations. As illustrated in Figure 1, small values of ϵ yield high nominal performance but weak robustness, whereas large values guarantee robustness but lead to overly pessimistic value estimates that can slow or destabilize learning. Thus, training policies with a large fixed ϵ forces the agent to optimize against a severely depressed value function, while training with small ϵ produces policies that are insufficiently robust at deployment. This inherent trade-off motivates the use of curriculum learning in DRRL, where we treat the robustness budget ϵ as the context of a curriculum: beginning with manageable uncertainty set and gradually expanding it as the agent's performance and robustness increases.

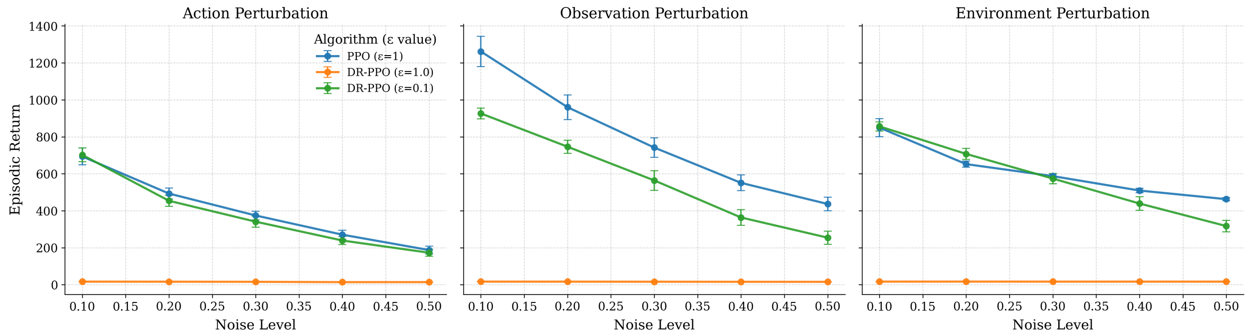


Figure 1: Robustness evaluation of DRPPO under action, observation, and environment perturbations for the Hopper environment. Each panel shows episodic returns as a function of noise levels. Policies trained with smaller fixed robustness budgets (ϵ) fail to handle perturbations effectively, while larger ϵ values produce overly conservative behavior, leading to suboptimal policies. This highlights the tradeoff between robustness and nominal performance for fixed ϵ settings.

1.1 Challenges and Contributions

The main challenge of training an effective DRRL policy is to automatically schedule the robustness budget ϵ so that an agent can efficiently learn a policy guaranteed to maintain performance under any environment uncertainty. Designing such a schedule is non-trivial: the task difficulty parameterized by ϵ lacks an intuitive or direct link to semantic properties of the RL task, and poor scheduling either undermines robustness or slows and destabilizes training. This motivates an approach in which the agent itself determines how quickly to expand the uncertainty set based on its learning progress and robustness level.

To address this problem, we introduce **Distributionally Robust Self-Paced Curriculum Reinforcement Learning (DR-SPCRL)**, a novel algorithm that automates curriculum generation for ϵ. We derive DR-SPCRL by instantiating a general curriculum learning framework for the specific mathematical structure of DRRL, with the novelty that the curriculum update is guided directly by the agent's current robustness to environmental uncertainty, enabling it to adaptively balance robustness and performance compared to heuristic-based approaches. By applying the Envelope Theorem to the primal DRRL problem, we formally show that the gradient of the robust value function with respect to the curriculum parameter ϵ is equal to the negative of the optimal dual variable, β^* [4]. This dual variable represents the marginal cost of robustness, a theoretically grounded measure of how much the agent is struggling at its current robustness level. DR-SPCRL leverages this signal to create an adaptive update rule that automatically balances the agent's competence with the need to progress the curriculum.

In summary, our main contributions are:

- We are the first to motivate and formalize the scheduling of the robustness budget ϵ in DRRL as a continuous, contextual curriculum learning problem to improve the stability of training DRRL policies.
- We introduce DR-SPCRL, a novel automated curriculum algorithm that leverages the dual structure of distributionally robust reinforcement learning (DRRL) to adaptively adjust the robustness budget ϵ based on the agent's learning and robustness progress.
- Empirical evaluations across diverse continuous control environments show that DR-SPCRL stabilizes training and achieves superior robustness–performance trade-offs, yielding an average 11.8% improvement in episodic returns compared to both non-robust baselines and DRRL agents trained with fixed or heuristic robustness schedules, while also achieving approximately $1.9\times$ the performance of the corresponding nominal RL algorithms.

2 Related Work

2.1 Curriculum Reinforcement Learning

Curriculum Learning [1] enhances generalization by structuring agent training from simpler to more complex tasks. Early approaches relied on manually ordered tasks or easily parameterized task sets [2, 5]. Modern methods automate this process, such as Self-Paced Contextual RL [6], which formulates the problem as an active optimization of an intermediate context distribution, balancing local reward maximization with a KL-divergence penalty to ensure progress towards a target distribution. This idea is grounded in information-theoretic frameworks like contextual relative entropy policy search. Other automated methods include using intrinsic motivation [7], generative models for environment [8, 9], and self-play [10]. However, the key limitation of these existing CRL methods is that they are specific to a narrow class of contexts, such as predefined semantic tasks or goal-conditioned settings, and offer no systematic mechanism or guarantees for handling broader or adversarial environmental variations. This motivates our work on Distributionally Robust Self-Paced Curriculum Reinforcement Learning (DR-SPCRL), which explicitly integrates robustness into the curriculum design so that the resulting policies maintain strong performance even under significant deviations from the training conditions.

2.2 Distributionally Robust Reinforcement Learning

Distributionally Robust Optimization [11, 12] refers to optimization where underlying parameters are unknown. For Reinforcement Learning [3, 13], this is formalized as a robust Markov Decision Process [14, 15]. Existing literature on DRRL focuses on extending RL methods such as Q-learning [13, 16], value iteration [17], and policy gradient methods [18, 19, 20, 21]. Additional research has focused on the model-based [22] and offline setting [23, 24, 25]. However, DRRL is largely limited to the tabular setting [17, 13, 26]. Recently, some works [27] have extended DRRL to the continuous setting by leveraging the closed form of the KL divergence uncertainty set. However, the primary pitfall of existing deep DRRL methods is their reliance on a fixed robustness budget ϵ. These methods lack an adaptive mechanism to manage this difficulty parameter during training, which can lead to the instability or overly conservative policies that our work aims to resolve.

3 Problem Formulation

We formalize the Distributionally Robust Self-Paced Curriculum Reinforcement Learning (DR-SPCRL) problem by starting from the standard Markov Decision Process (MDP)

$$M = (\mathcal{S}, \mathcal{A}, \gamma, P, r)$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, γ is the discount factor, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition kernel, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. A policy π_θ seeks to maximize the expected return

$$J(\pi_\theta, P) = \mathbb{E}_{\pi_\theta, P}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\Big].$$

In many applications the true transition kernel is not exactly known; instead, we assume it lies in an *uncertainty set* $\mathcal{P}(\epsilon)$ around a nominal model P_0:

$$\mathcal{P}(\epsilon) = \{P : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \mid D(P(\cdot|s,a), P_0(\cdot|s,a)) \leq \epsilon, \forall s, a\} \tag{1}$$

where D is a divergence measure and $\epsilon \geq 0$ is the *robustness budget*. The resulting robust MDP objective is

$$\max_\theta V_{\text{robust}}(\pi_\theta; \epsilon), \qquad V_{\text{robust}}(\pi_\theta; \epsilon) = \inf_{P \in \mathcal{P}(\epsilon)} J(\pi_\theta, P).$$

To model multiple robustness levels within a single framework, we view the robustness budget as a *context variable* $c \in C$ with

$$C = [0, \epsilon_{\text{budget}}], \qquad c \equiv \epsilon.$$

Each context specifies a different uncertainty set $\mathcal{P}(c)$ and thus a distinct robust control problem, yielding a contextual robust MDP. Rather than training directly at the most challenging context $\epsilon_{\max}$, distributionally robust curriculum reinforcement learning organizes learning through a curriculum of increasing budgets

$$\mathcal{E} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_T), \qquad 0 \leq \epsilon_t \leq \epsilon_{\text{budget}}, \quad \epsilon_{t+1} \geq \epsilon_t,$$

where each ϵ_{t+1} represents a more difficult level of uncertainty than ϵ_t. At stage t, the agent solves the robust MDP corresponding to ϵ_t:

$$\max_{\theta_t} V_{\text{robust}}(\pi_{\theta_t}; \epsilon_t), \quad t = 1, \ldots, T,$$

and the budget is increased only when sufficient progress is achieved. Our objective is therefore to jointly learn a sequence of policies $\{\pi_{\theta_t}\}_{t=1}^T$ and the curriculum $\mathcal{E}$ so that the final policy attains strong robust performance at the target uncertainty level ϵ_{budget}.

4 Method

In this section, we detail our proposed algorithm. We begin by establishing how DRRL can be used in the deep reinforcement learning setting. We then formally derive our curriculum reinforcement learning algorithm, which we term Distributionally Robust Self-Paced Curriculum Reinforcement Learning (DR-SPCRL), by starting with general curriculum reinforcement learning framework and integrating the distributionally robust formulation.

4.1 Deep Distributionally Robust RL

We consider a DRRL setting where the transition model P is unknown but belongs to a KL-divergence ball $\mathcal{P}(\epsilon)$ of radius ϵ around a nominal model P_0. The goal is to find a policy π_θ with parameters θ that maximizes the robust value function, V_{robust}, defined by the primal problem:

$$V = \inf_{P \in \{P \mid D_{KL}(P, P_0) \leq \epsilon\}} \mathbb{E}_{\pi_\theta, P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

Solving this infimum problem is generally intractable. However, as strong duality holds for the KL-divergence uncertainty set, this problem can be equivalently reformulated into its dual, which involves a maximization over the dual variable $\beta \geq 0$

$$V_{\text{robust}}(s, a) = \sup_{\beta \geq 0}\left(-\beta \log\left(\mathbb{E}_{s' \sim P_0(\cdot|s,a)}\left[\exp\left(-\frac{V(s')}{\beta}\right)\right]\right) - \beta\epsilon\right) \tag{2}$$

In our work, we approximate this optimal dual variable, $\beta^*(s, a)$, using a neural network $\beta_\phi(s, a)$. The agent's policy is then trained using any RL algorithm to estimate the value function, but with policy updates using the robust value function.

4.2 The DR-SPCRL Algorithm

We derive our curriculum generation method from the inference-based self-paced learning framework of [6], which poses curriculum generation as a joint optimization problem over the policy parameters θ and curriculum parameters ν:

$$\begin{aligned} \max_{\theta, \nu} \quad & \mathbb{E}_{p(c|\nu)}[J(\theta, c)] - \alpha D_{\text{KL}}(p(c|\nu) \,\|\, \mu(c)) \\ \text{s.t.} \quad & D_{\text{KL}}(p(c|\nu) \,\|\, p(c|\nu_0)) \leq \eta \end{aligned} \tag{3}$$

Here, $J(\theta, c)$ is the performance on task c, $\mu(c)$ is the target task distribution, α is a pacing parameter controlling exploration speed, and η is a trust region limiting large jumps between contexts. We can now frame our distributionally robust RL problem as a self-paced curriculum learning problem. We can define $J(\theta, c)$ as the robust value function V_{robust} with $c = \epsilon$, the curriculum itself as a probability distribution over ϵ and $p(\epsilon|\nu)$, and the target distribution $\mu(c)$ as a Dirac delta distribution at the target $\delta(\epsilon - \epsilon_{\text{budget}})$. Our goal is to derive a practical update rule for a single, continuous ϵ. We fix the policy θ and focus on the curriculum update step. The objective is to find the new curriculum distribution $p(\epsilon|\nu)$ that solves

$$\max_{p(\epsilon|\nu)} \quad \int p(\epsilon|\nu) V_{\text{robust}} d\epsilon - \alpha D(p(\epsilon|\nu) \,\|\, \mu(\epsilon)) \tag{4}$$

Because our target distribution $\mu(\epsilon)$ is a single point ($\delta(\epsilon - \epsilon_{\text{budget}})$), the KL-divergence term will strongly incentivize the optimal curriculum $p(\epsilon|\nu)$ to also concentrate its mass around a single point. For our practical implementation, we therefore model the curriculum as a single-point estimate $p(\epsilon|\nu) = \delta(\epsilon - \nu)$, where ν is the current value of the curriculum's ϵ.

With this model, the expectation $\mathbb{E}_{p(\epsilon|\nu)}[V_{\text{robust}}]$ simplifies to $V_{\text{robust}}(\theta; \nu)$. Additionally, the KL-divergence terms are not suitable for the Dirac delta functions, so we use the Euclidean distance between ϵ and ϵ_{budget} as an approximation. This transforms the distributional optimization problem into an equivalent optimization over the point-wise curriculum parameter. Thus, our curriculum learning optimization problem reduces to maximizing a constrained scalar objective using the current uncertainty radius:

$$\begin{aligned} \max_{\epsilon} \; & V_{\text{robust}}(\theta; \epsilon) - \alpha(\epsilon - \epsilon_{\text{budget}})^2 \\ \text{s.t.} \quad & (\epsilon - \epsilon_t)^2 \leq \eta \end{aligned} \tag{5}$$

To solve this constrained optimization problem, we form its Lagrangian. Let $f(\epsilon) = V_{\text{robust}}(\theta; \epsilon) - \alpha(\epsilon - \epsilon_{\text{budget}})^2$ be our objective. The Lagrangian for the curriculum optimization is:

$$\mathcal{L}_{\text{curr}}(\epsilon, \lambda) = f(\epsilon) - \lambda \left((\epsilon - \epsilon_t)^2 - \eta\right) \tag{6}$$

Here, $\lambda \geq 0$ is the Lagrange multiplier for the trust-region constraint. Setting the derivative of $\mathcal{L}_{\text{curr}}$ with respect to ϵ to zero gives the optimality condition:

$$\nabla_\epsilon \mathcal{L}_{\text{curr}} = \nabla_\epsilon f(\epsilon) - 2\lambda(\epsilon - \epsilon_t) = 0 \tag{7}$$

However, the primary challenge in computing $\nabla_\epsilon \mathcal{L}_{\text{curr}}$ is that V_{robust} is defined implicitly via an inner minimization over $\mathcal{P}$. Directly differentiating through $P^*(\epsilon)$ (the worst case transition kernel for the current policy) would require backpropagating through a very high-dimensional constrained optimization problem. To circumvent this, we can leverage the Envelope Theorem [4], which states that any optimization problem of the form

$$g(\epsilon) = \max_{x \in X} f(x, \epsilon)$$

then under mild regularity conditions:

$$\nabla_\epsilon g(\epsilon) = \left.\frac{\partial f(x, \epsilon)}{\partial \epsilon}\right|_{x = x^*(\epsilon)}$$

The Lagrangian for V_{robust} is $\mathcal{L}(P, \beta) = V + \beta(D_{KL}(P \parallel P_0) - \epsilon)$ and the partial derivative of the Lagrangian with respect to ϵ is $-\beta$. By the Envelope Theorem, the derivative of the value function V_{robust} is the partial derivative of the Lagrangian evaluated at the worst case transition kernel (P^*) and associated worst case dual variable (β^*):

$$\nabla_\epsilon V_{\text{robust}} = \left.\frac{\partial \mathcal{L}(P, \beta)}{\partial \epsilon}\right|_{P = P^*, \beta = \beta^*} = -\beta^*(\epsilon) \tag{8}$$

This formally shows that the gradient of the robust value function is the negative of the optimal dual variable β^*, which represents the marginal cost of robustness. Substituting this into the optimality condition, we have:

$$-\beta^*(\epsilon) - 2\alpha(\epsilon - \epsilon_{\text{budget}}) - 2\lambda(\epsilon - \epsilon_t) = 0 \tag{9}$$

Solving for ϵ and substituting $\epsilon = \epsilon_{t+1}$ yields an implicit equation for the optimal next step. To create a practical, explicit update rule, we make a first-order approximation by evaluating the gradient at ϵ_t and treating the step-size term $\frac{1}{2(\alpha+\lambda)}$ as an effective curriculum learning rate, denoted λ_{curr}. This learning rate now implicitly enforces the trust-region constraint η, leading to the final update rule:

$$\epsilon_{t+1} = \epsilon_t - \lambda_{\text{curr}} \left(\beta^*(\epsilon_t) + 2\alpha(\epsilon_t - \epsilon_{\text{budget}})\right) \tag{10}$$

4.3 Practical Implementation

In practice, we do not have access to β^*, so we use β_ϕ as an approximation with ϕ as a neural network updated to maximize V_{robust}. Our final algorithm is RL-algorithm agnostic and proceeds in a block-coordinate ascent fashion, similar to [6]. At each timestep, we compute the the current curriculum budget ϵ_t and the policy π_θ and β_ϕ are updated. This step is RL algorithm-agnostic, and any algorithm that learns the value function or can be used to approximate the value function can be made robust by using V_{robust} in the update and updating ϕ in the direction that maximizes V_{robust}. We then compute the curriculum gradient with respect to ϵ_t and transition to ϵ_{t+1}. We give the full algorithm in Algorithm 1.

5 Experiments

We conduct a comprehensive set of experiments to empirically validate our proposed Distributionally Robust Self-Paced Curriculum Reinforcement Learning (DR-SPCRL) algorithm. Our evaluation targets three key hypotheses: (1) DR-SPCRL produces policies with greater robustness than standard non-robust baselines; (2) its adaptive curriculum yields more stable training and better final performance than robust training with a fixed or heuristic schedule; and (3) the method is general-purpose, providing benefits across diverse algorithms and domains. To demonstrate this, we integrate DR-SPCRL with three state-of-the-art deep RL algorithms representing distinct paradigms: Proximal Policy Optimization (PPO) [28], Soft Actor-Critic [29] (SAC), and Deep Deterministic Policy Gradients (DDPG) [30].

We evaluate our algorithm on the Hopper, Humanoid, Half-Cheetah, and Walker2d continuous control environments. We assess policy robustness using two primary metrics: nominal performance, which is the average episodic return

Algorithm 1 Distributionally Robust Self-Paced Curriculum Reinforcement Learning (DR-SPCRL)

Require: discount factor γ, initial budget ϵ_{start}, final budget ϵ_{budget}, pacing parameter α, curriculum learning rate λ_{curr}, number of iterations T, rollout steps per iteration N, and number of critic updates K.
Initialize policy π_θ, dual model β_ϕ, curriculum $\epsilon_t \leftarrow \epsilon_{\text{start}}$, and experience buffer $\mathcal{D}$.
for $t = 1$ to T **do**
    Collect N steps of experience (s, a, r, s') using π_θ; store in $\mathcal{D}$.
    for $k = 1$ to K **do**
        Sample minibatch $\{(s_j, a_j, r_j, s'_j)\}$ from $\mathcal{D}$.
        Compute dual loss $L(\phi) = -V_{\text{robust}}(\theta; \epsilon_t)$ from Eq. (2).
        Update β_ϕ via gradient descent on $L(\phi)$.
    end for
    Compute robust value and advantages using updated β_ϕ.
    Update policy parameters θ via RL algorithm with robust value and advantage functions.
    Estimate $\beta^*(\epsilon_t) \approx \mathbb{E}_{(s,a)\sim\mathcal{B}}[\beta_\phi(s, a)]$ from minibatch $\mathcal{B}$.
    Compute regularization gradient $g_{\text{reg}} = 2\alpha(\epsilon_t - \epsilon_{\text{budget}})$ from Eq. (7).
    Update curriculum using Eq. (10): $\epsilon_{t+1} \leftarrow \epsilon_t - \lambda_{\text{curr}}(\beta_\phi + g_{\text{reg}})$.
    Project ϵ_{t+1} into $[0, \epsilon_{\text{budget}}]$.
end for
return final policy π_θ.

in the unperturbed training environment, and robust performance, which is the average episodic return under a suite of held-out perturbations not encountered during training. This evaluation suite is designed to simulate a range of real-world uncertainties and is composed of three distinct and formally defined perturbation types:

First, to simulate sensor noise, we apply observation noise. Formally, at each timestep t, the agent receives a perturbed state observation s'_t instead of the true state s_t. The perturbed state is given by:

$$s'_t = s_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma_{obs}^2 I)$$

Here, η_t is a noise vector sampled from a zero-mean Gaussian distribution with a fixed standard deviation σ_{obs}, applied independently to each dimension of the state vector.

Second, to simulate actuator imprecision or faults, we apply action noise. Specifically, the action a_t selected by the policy is replaced by a perturbed action a'_t before being executed in the environment. This replacement occurs with a fixed probability p_{act}:

$$a'_t = \begin{cases} a_{\text{rand}} \sim U(\mathcal{A}) & p_{\text{act}} \\ \pi_t(s_t) & 1 - p_{\text{act}} \end{cases}$$

Third, to simulate sim-to-real gaps or systemic changes in the environment itself, we apply environment noise. This is achieved by modifying the underlying physical parameters of the MuJoCo simulation at the start of each evaluation episode. Let ξ_0 be the vector of nominal physics parameters (e.g., body masses, friction coefficients, joint damping). For each evaluation episode, we sample a new parameter vector ξ' where each component ξ'_i is drawn from a uniform distribution centered around its nominal value:

$$\xi'_i = U(1 - \delta_{\text{env}}, 1 + \delta_{\text{env}})\xi'_0$$

where δ_{env} defines the percentage range of the perturbation. The episode is then run entirely within the simulator configured with these perturbed physics ξ'. For our experiments, we only perturb the body mass, the damping force, and the friction of the MuJoCo environments.

We compare our DRSPCRL algorithm to five curriculum reinforcement learning algorithms where we set the context as the epsilon radius of the uncertainty set:

- **Vanilla** The standard, non-robust RL algorithm, which corresponds to training with $\epsilon = 0$. This serves as our baseline for nominal performance and demonstrates the fragility of standard policies.
- **Fixed Budget** Distributionally Robust RL trained with a fixed, high robustness budget throughout training. This baseline is designed to highlight the instability or over-conservatism that arises from training without a curriculum.
- **Linear Schedule** A baseline that employs a handcrafted curriculum where the robustness budget ϵ is linearly annealed from 0 to the final target budget over the course of the training run.

- **ACCEL** An adaptation of the regret-based curriculum generation method, ACCEL [31]. We maintain a replay buffer of ϵ values, where the regret score of an ϵ is defined by the mean dual variable β^*. The curriculum progresses by sampling and editing high-regret ϵ values from this buffer.
- **SPACE** An adaptation of the Self-Paced Contextual Evaluation (SPACE) algorithm [32]. This method employs a heuristic where the curriculum budget ϵ is increased only when the agent's robust value function has plateaued, indicating that the agent has mastered the current difficulty level.

For all experiments we set the learning rate for the ϕ network to be 5×10^{-4} and the number of steps K to be 5. We also set the ϵ_{budget} to be 1, which under the KL divergence captures a wide range of possible transition kernel perturbations. We evaluate each policy after training for 1 million steps for 100 episodes, and report the mean and the standard error for a 95% confidence interval. All experiments were conducted on an RTX 4080 Laptop GPU with 4 i9-13900HX CPUs.

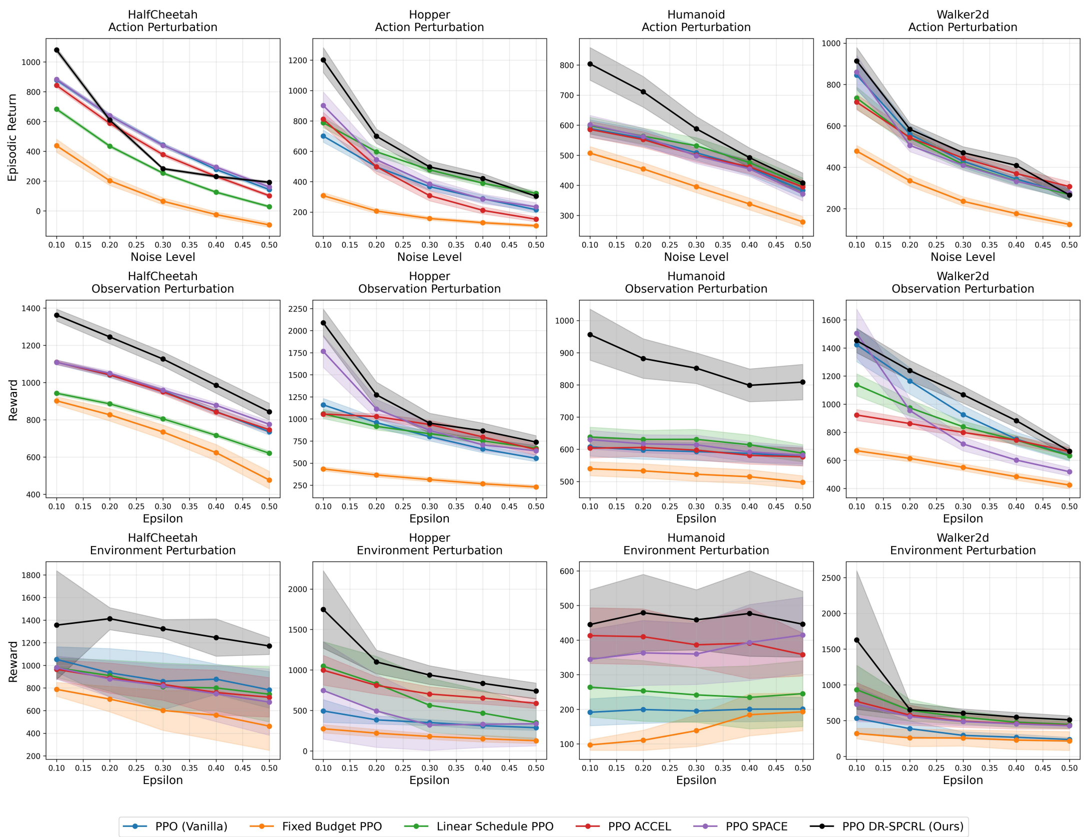


Figure 2: PPO Robustness under observation, action and environmental perturbations.

5.1 Results

DR-SPCRL consistently achieves higher robustness-performance trade-offs. Across Figures 2-4, DR-SPCRL maintains higher episodic returns under strong perturbations while retaining competitive nominal performance. For PPO under environmental perturbations (Figure 2, bottom row), DR-SPCRL achieves an episodic return of approximately 2000 at the maximum perturbation, outperforming Fixed Budget ($\sim$1000) and Vanilla PPO (near-zero). SAC results (Figure 3, right) show DR-SPCRL at an episodic return of $\sim$2500 versus Linear Schedule ($\sim$1500) and Fixed Budget ($\sim$2000), while DDPG exhibits similar improvements (Figure 4). This superior performance arises from DR-SPCRL's dual-variable guided curriculum, which adapts the robustness budget ϵ according to the agent's learning progress. By starting with small ϵ, the agent establishes a strong nominal policy, and as the dual variable β^* signals mastery, ϵ gradually increases, systematically exposing the agent to higher uncertainty. This progressive exposure prevents the

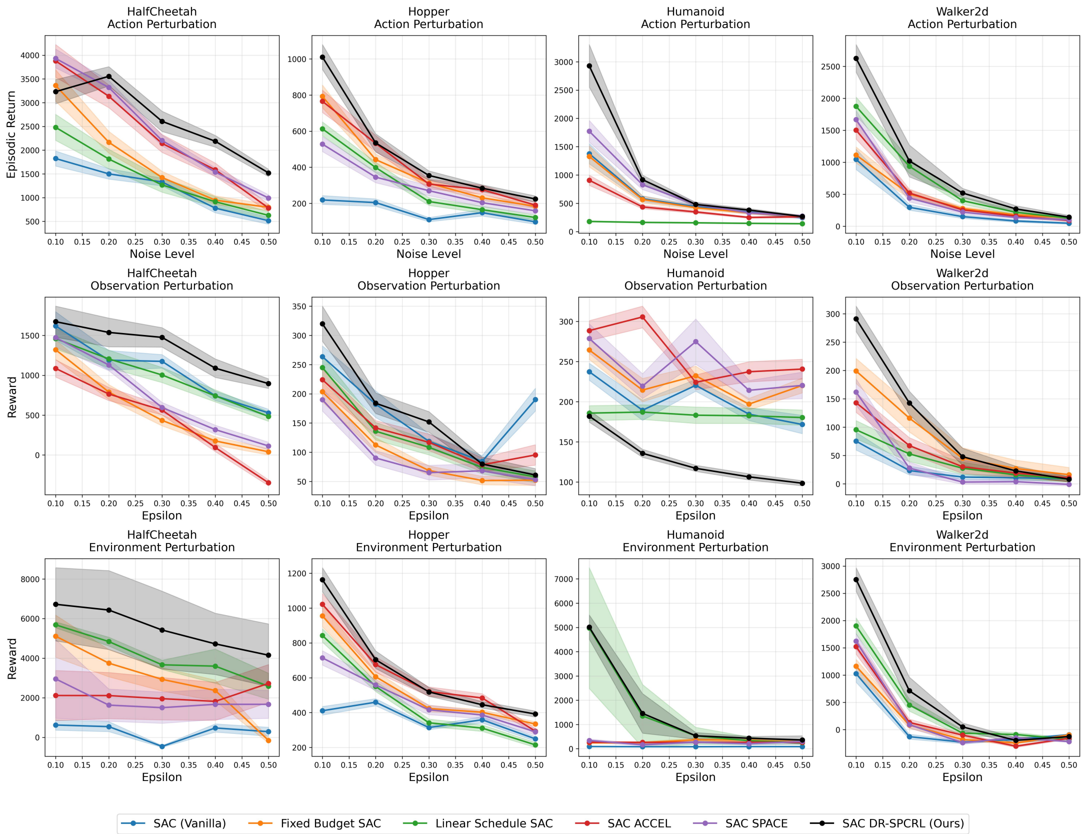


Figure 3: SAC Robustness under observation, action and environmental perturbations.

collapse under perturbation seen in Fixed Budget approaches and avoids the instability of heuristic schedules, enabling DR-SPCRL to consistently balance robustness and performance across algorithms and perturbation types. In total, DR-SPCRL is able to outperform fixed or heuristic based CRL algorithms by 11.8% across action, observation, and environment perturbations. It also improves the robustness over the nominal RL algorithm by 1.9×.

Fixed robustness budgets force an unavoidable compromise between nominal and robust performance. Fixed-ϵ approaches demonstrate a clear trade-off between nominal performance and robustness. In PPO observation noise experiments (Figure 2), Fixed Budget ($\epsilon = 1.0$) maintains higher returns under strong noise but sacrifices nominal performance, achieving only $\sim$3000 at zero noise compared to Vanilla PPO's $\sim$3500. Conversely, Vanilla PPO ($\epsilon = 0$) preserves nominal performance but collapses under high noise, dropping to $\sim$500 at maximum perturbation while Fixed Budget maintains $\sim$1800. DDPG shows a similar pattern (Figure 4): Fixed Budget preserves an episodic return of $\sim$1000 under maximum action perturbations but only $\sim$2500 under small perturbations versus Vanilla's $\sim$3500. This occurs because a static robustness objective cannot adjust to the agent's evolving capabilities: small ϵ ignores robustness, while large ϵ depresses nominal returns, creating an unavoidable performance-robustness compromise that adaptive curricula like DR-SPCRL can resolve.

Heuristic curriculum methods provide inconsistent robustness benefits across environments Heuristic approaches, including Linear Schedule, SPACE, and ACCEL, yield performance that is highly dependent on the algorithm and perturbation type. For SAC environmental perturbations (Figure 3), Linear Schedule and SPACE degrade rapidly beyond a noise level of 0.3, dropping to $\sim$1500 and $\sim$1000, respectively, while DR-SPCRL maintains an episodic return of $\sim$2500. In DDPG with action perturbations (Figure 4, top row), ACCEL collapses to near-zero by a noise level of 0.5, whereas DR-SPCRL preserves an episodic return of $\sim$1500. These inconsistencies arise because heuristic methods rely on predefined timesteps or plateau detection rather than a principled measure of learning progress. They may advance too quickly, overwhelming the agent, or too slowly, failing to develop full robustness. In contrast, DR-SPCRL

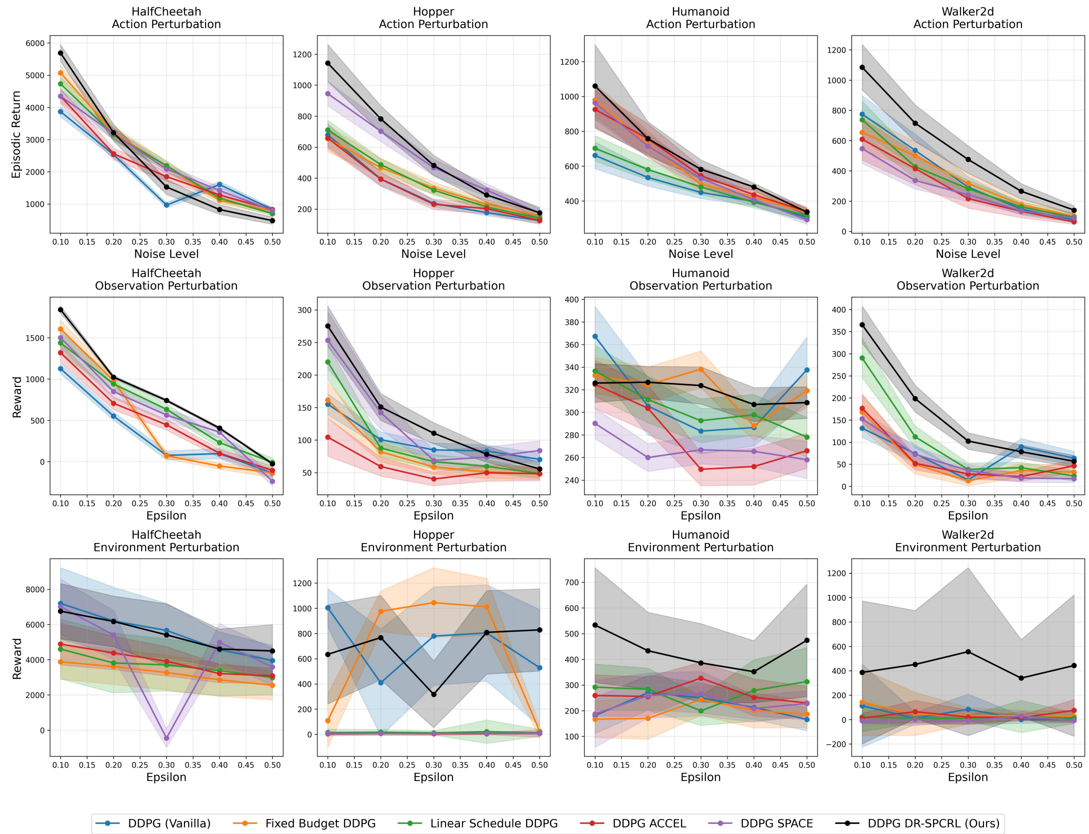


Figure 4: DDPG Robustness under observation, action and environmental perturbations.

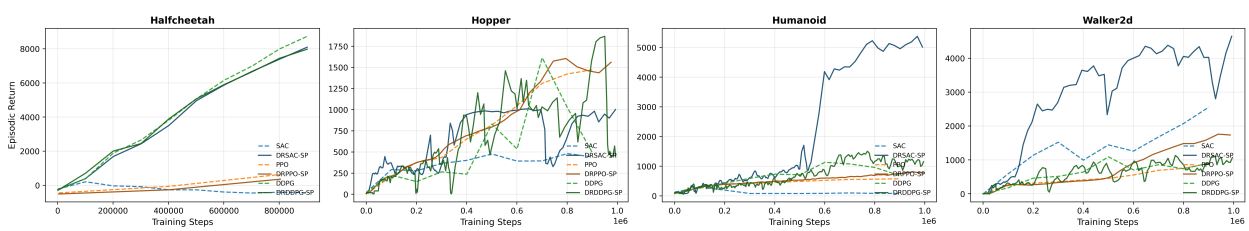
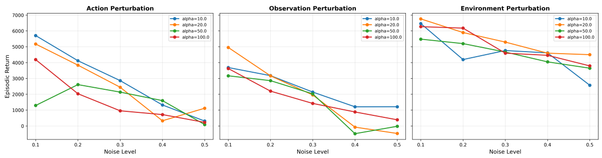


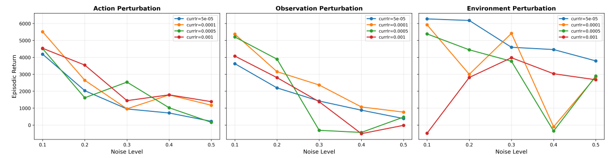
Figure 5: Comparison of training curves across the four continuous-control environments under different curriculum strategies. DR-SPCRL demonstrates smoother and more stable learning dynamics compared to the non-robust baselines, often significantly increasing the nominal reward.

leverages the dual variable β^* to adaptively scale task difficulty based on actual agent performance, yielding consistent robustness gains across algorithms and perturbation types.

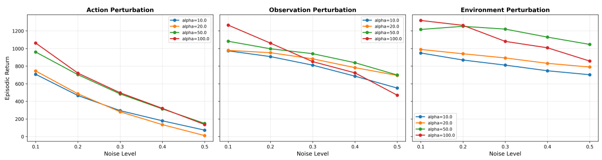
Environmental perturbations highlight the advantage of adaptive curricula Environmental perturbations, which alter fundamental dynamics such as mass, friction, and damping, present the greatest challenge for robustness because standard RL training assumes consistent environment parameters, even as observations and actions vary during the trajectory. PPO results (Figure 2) show DR-SPCRL maintaining an episodic return of $\sim$2000 at maximum perturbation, compared to Fixed Budget's $\sim$1000, while DDPG (Figure 4) preserves a return of $\sim$1500 versus Linear Schedule's $\sim$800 and Fixed Budget's $\sim$600. Fixed budgets either over-constrain learning or fail to sufficiently challenge the agent, and heuristics cannot reliably adjust to non-linear, environment-specific difficulty progressions. DR-SPCRL's



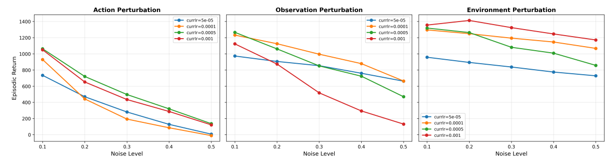
(a) DDPG: Varying pacing parameters α



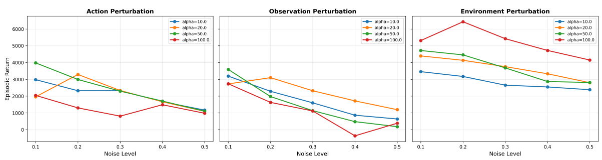
(b) DDPG: Varying curriculum learning rates



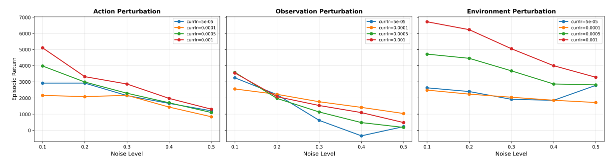
(c) PPO: Varying pacing parameters α



(d) PPO: Varying curriculum learning rates



(e) SAC: Varying pacing parameters α



(f) SAC: Varying curriculum learning rates

Figure 6: Ablation studies evaluating DR-SPCRL's robustness under fixed hyperparameters. Each panel shows episodic returns under observation, action, and environmental perturbations. Left column: results with fixed pacing parameters α. Right column: results with fixed curriculum learning rates. Across all algorithms (DDPG, PPO, SAC), DR-SPCRL maintains high robustness and demonstrates stability despite suboptimal hyperparameter choices.

dual-variable guided curriculum automatically scales the robustness budget based on observed agent proficiency, exposing the policy to systematically increasing dynamics changes. This capability enables effective learning under substantial environmental variation, highlighting DR-SPCRL's potential for sim-to-real transfer and other domains with systematic environmental uncertainty.

5.2 Training Results

DR-SPCRL consistently improves training performance across environments and algorithms Across all tested environments and base algorithms, DR-SPCRL variants (denoted with "-SP") demonstrate higher episodic returns and smoother learning trajectories compared to standard SAC, PPO, and DDPG. In HalfCheetah, DRSAC-SP reaches nearly 8000 episodic return, substantially exceeding standard SAC, while in Humanoid, DRPPO-SP achieves around 5000, roughly double standard PPO's performance. These improvements are evident throughout training, not just at the final evaluation, with DR-SPCRL agents avoiding the plateaus, instabilities, and sudden performance collapses frequently observed in non-robust RL training, particularly in high-dimensional or challenging control tasks.

Adaptive curriculum and dual-variable guidance explain DR-SPCRL's superior training behavior The key to DR-SPCRL's success during training is its dual-variable guided curriculum, which treats the robustness budget ϵ as a dynamic, learnable parameter. Early in training, the agent faces small uncertainty sets, enabling it to establish strong nominal policies without being overwhelmed. As the agent demonstrates proficiency, the dual variable β^* automatically signals when the curriculum can safely increase ϵ, progressively introducing harder perturbations. This ensures the policy is continually challenged but not overstressed, preventing collapse under large perturbations and avoiding overfitting to nominal dynamics. Standard RL lacks this mechanism, optimizing only for a fixed environment and consequently struggling with stability and robustness, which explains the smoother, higher-performing training curves observed with DR-SPCRL.

5.3 Ablation Study

To assess the robustness of DR-SPCRL under varied conditions, we conducted ablation experiments by varying either the pacing parameter α or the curriculum learning rate for the entire training run (Figure 6). Across DDPG, PPO, and SAC, the results demonstrate that DR-SPCRL consistently maintains high episodic returns under observation, action, and environmental perturbations, even with suboptimal hyperparameter choices.

For DDPG with fixed α values (Figure 6a), all configurations sustain returns above 1500 under observation perturbations, with minimal differences between $\alpha = 0.1$ ($\sim$1800) and $\alpha = 1.0$ ($\sim$1700). Similarly, varying the curriculum learning rate (Figure 6b) yields comparable performance across all perturbation types. PPO exhibits similar stability: environmental perturbation returns (Figures 6c and 6d) vary by less than 10% across different hyperparameter settings, while SAC also shows robust performance under both fixed α and learning rates (Figures 6e and 6f).

This stability can be attributed to DR-SPCRL's self-correcting curriculum mechanism. The dual variable β^* dynamically adjusts the robustness budget based on the agent's current capability, naturally compensating for suboptimal hyperparameters and ensuring that the curriculum progresses at an appropriate rate. As a result, the algorithm maintains effective robustness across all noise levels without requiring extensive hyperparameter tuning. Even when performance is slightly reduced due to certain fixed values, the overall robustness and learning behavior remain largely preserved, demonstrating DR-SPCRL's practical applicability in real-world scenarios where exhaustive hyperparameter optimization may be infeasible.

6 Conclusion and Future Work

In this paper, we introduce DR-SPCRL, a novel curriculum learning approach for distributionally robust reinforcement learning. Our method applies self-paced curriculum learning by treating the ϵ-radius of the uncertainty set as the context for each training iteration. Experimental results across three baseline RL algorithms, 5 CRL algorithms, and four environments demonstrate that DR-SPCRL improves robustness and outperforms existing curriculum reinforcement learning methods.

Although DR-SPCRL is designed for single-agent settings, there is growing interest in multi-agent distributionally robust algorithms. Our framework could be extended to the multi-agent setting by employing a joint robust value function and appropriately adapting the formulation and algorithm. Future work could also explore integrating DR-SPCRL with model-based RL or planning methods, potentially allowing the agent to anticipate perturbations more efficiently and further improve robustness.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[2] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

[3] Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. Distributionally robust reinforcement learning. *arXiv preprint arXiv:1902.08708*, 2019.

[4] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

[5] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey, 2022.

[6] Pascal Klink, Carlo D'Eramo, Jan R Peters, and Joni Pajarinen. Self-paced deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:9216–9227, 2020.

[7] Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play, 2018.

[8] Anirudh Satheesh, Keenan Powell, and Hua Wei. cmalc-d: Contextual multi-agent llm-guided curriculum learning with diversity-based context blending. *arXiv preprint arXiv:2508.20818*, 2025.

[9] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents, 2018.

[10] Yuqing Du, Pieter Abbeel, and Aditya Grover. It takes four to tango: Multiagent selfplay for automatic curriculum generation, 2022.

[11] Daniel Kuhn, Soroosh Shafiee, and Wolfram Wiesemann. Distributionally robust optimization, 2025.

[12] Peter J. Huber. *Robust Statistics*. Wiley, 1981.

[13] Zijian Liu, Qinxun Bai, Jose Blanchet, Perry Dong, Wei Xu, Zhengqing Zhou, and Zhengyuan Zhou. Distributionally robust q-learning. In *International Conference on Machine Learning*, pages 13623–13643. PMLR, 2022.

[14] Garud Iyengar. Robust dynamic programming. *Math. Oper. Res.*, 30:257–280, 05 2005.

[15] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

[16] Shengbo Wang, Nian Si, Jose Blanchet, and Zhengyuan Zhou. Sample complexity of variance-reduced distributionally robust q-learning. *Journal of Machine Learning Research*, 25(341):1–77, 2024.

[17] Miao Lu, Han Zhong, Tong Zhang, and Jose Blanchet. Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithms. *Advances in Neural Information Processing Systems*, 37:12528–12580, 2024.

[18] Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *International conference on machine learning*, pages 23484–23526. PMLR, 2022.

[19] Navdeep Kumar, Esther Derman, Matthieu Geist, Kfir Y Levy, and Shie Mannor. Policy gradient for rectangular robust markov decision processes. *Advances in Neural Information Processing Systems*, 36:59477–59501, 2023.

[20] Ruida Zhou, Tao Liu, Min Cheng, Dileep Kalathil, PR Kumar, and Chao Tian. Natural actor-critic for robust reinforcement learning with function approximation. *Advances in neural information processing systems*, 36:97–133, 2023.

[21] Qiuhao Wang, Chin Pang Ho, and Marek Petrik. Policy gradient in robust mdps with global convergence guarantee. In *International Conference on Machine Learning*, pages 35763–35797. PMLR, 2023.

[22] Shyam Sundhar Ramesh, Pier Giuseppe Sessa, Yifan Hu, Andreas Krause, and Ilija Bogunovic. Distributionally robust model-based reinforcement learning with large state spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 100–108. PMLR, 2024.

[23] Xiaoteng Ma, Zhipeng Liang, Jose Blanchet, Mingwen Liu, Li Xia, Jiheng Zhang, Qianchuan Zhao, and Zhengyuan Zhou. Distributionally robust offline reinforcement learning with linear function approximation. *arXiv preprint arXiv:2209.06620*, 2022.

[24] Jose Blanchet, Miao Lu, Tong Zhang, and Han Zhong. Double pessimism is provably efficient for distributionally robust offline reinforcement learning: Generic algorithm and robust partial coverage. *Advances in Neural Information Processing Systems*, 36:66845–66859, 2023.

[25] Zhishuai Liu and Pan Xu. Minimax optimal and computationally efficient algorithms for distributionally robust offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:86602–86654, 2024.

[26] Pierre Clavier, Laixi Shi, Erwan Le Pennec, Eric Mazumdar, Adam Wierman, and Matthieu Geist. Near-optimal distributionally robust reinforcement learning with general l_p norms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 1750–1810. Curran Associates, Inc., 2024.

[27] Mingxuan Cui, Duo Zhou, Yuxuan Han, Grani A Hanasusanto, Qiong Wang, Huan Zhang, and Zhengyuan Zhou. Dr-sac: Distributionally robust soft actor-critic for reinforcement learning under uncertainty. *arXiv preprint arXiv:2506.12622*, 2025.

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[29] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[30] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[31] Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, pages 17473–17498. PMLR, 2022.

[32] Theresa Eimer, André Biedenkapp, Frank Hutter, and Marius Lindauer. Self-paced context evaluation for contextual reinforcement learning. In *International Conference on Machine Learning*, pages 2948–2958. PMLR, 2021.

[33] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

A Appendix

A.1 Algorithm Hyperparameters

Our Hyperparameters for PPO, SAC, and DDPG are taken from the default CleanRL [33] implementations for these algorithms. These are listed in Tables 1 (PPO), 2 (SAC), and 3 (DDPG). These hyperparameters are also used for the equivalent Distributionally Robust versions. We also list hyperparameters for the SPACE, ACCEL, and Linear Schedule comparison curriculum learning environments in Tables 4 (SPACE), 5 (ACCEL), and 6 (Linear Schedule).

Table 1: Hyperparameters for PPO

PPO Hyperparameters	
Eps Clip	0.2
Learning Rate	0.0003
Discount Factor (γ)	0.99
GAE Lambda	0.95
Hidden Dim	64
Num Steps	2048
Mini Epochs	10
Num Minibatches	32
Entropy Coef	0.0
Value Function Coef	0.5
Max Grad Norm	0.5
Normalize Advantages	True
Clip Value Loss	True

Table 2: Hyperparameters for SAC

SAC Hyperparameters	
Buffer Size	1000000
Batch Size	256
Learning Starts	5000
Discount Factor (γ)	0.99
Tau (τ)	0.005
Policy Learning Rate	0.0003
Q Learning Rate	0.001
Policy Frequency	2
Target Network Frequency	1
Alpha (α)	0.2
Autotune	True
Hidden Dim (Actor)	256
Hidden Dim (Q-Network)	256

Table 3: Hyperparameters for DDPG

DDPG Hyperparameters	
Buffer Size	1000000
Batch Size	256
Learning Starts	25000
Discount Factor (γ)	0.99
Tau (τ)	0.005
Learning Rate	0.0003
Policy Frequency	2
Exploration Noise	0.1
Hidden Dim (Actor)	256
Hidden Dim (Q-Network)	256

Table 4: Hyperparameters for SPACE across all algorithms

SPACE Hyperparameters	PPO	SAC	DDPG
Eps Start	0	0	0
Eps Step	0.01	0.01	0.01
Curriculum Interval	10	100	1000
Curriculum Start	5	1000	50000
Plateau Threshold	0.1	0.1	0.1

Table 5: Hyperparameters for ACCEL across all algorithms

ACCEL Hyperparameters	
Eps Buffer	50
Replay Prob	0.8
Editor Noise std	0.01
Generator Range	0.02
Curriculum Interaval	100

Table 6: Hyperparameters for Linear Schedule across all algorithms

Linear Schedule Hyperparameters	PPO	SAC	DDPG
Eps Start	0	0	0
Eps Step	0.01	0.01	0.01
Curriculum Start	5	5	50000