Robust Situational Reinforcement Learning in Face of Context Disturbances

Jinpeng Zhang [*1] Yufeng Zheng [2] Chuheng Zhang [3] Li Zhao [3] Lei Song [3] Yuan Zhou [4] Jiang Bian [3]

Abstract

In many real-world tasks, the presence of dynamic and uncontrollable environmental factors, commonly referred to as *context*, plays a crucial role in the decision-making process. Examples of such factors include customer demand in inventory control and the speed of the lead car in autonomous driving. One of the challenges of reinforcement learning in these applications is that the true context transitions can be easily exposed to some unknown source of contamination, leading to *a shift of context transitions* between source domains and target domains, which could cause performance degradation for RL algorithms. To tackle this problem, we propose the robust situational Markov decision process (RS-MDP) framework which captures the possible deviations of context transitions explicitly. To scale to large context space, we introduce the softmin smoothed robust Bellman operator to learn the robust Q-value approximately, and extend existing RL algorithm SAC to learn the desired robust policies under our RS-MDP framework. We conduct experiments on several locomotion tasks with dynamic contexts and inventory control tasks to demonstrate that our algorithm can generalize better and be more robust against context disturbances, and outperform existing basic RL algorithms that do not consider robustness and robust RL algorithms that consider robustness over the whole state transitions.

1. Introduction

In many real-world applications, there are dynamic environmental factors, which cannot be influenced by agents' actions, but is vital for agents' decision-making, e.g., the speed of lead cars in autonomous driving, customer demand in inventory control, stock price in optimized trade execution, etc. We refer to such environmental factors as *contexts* for simplicity (also called exogenous states (Dietterich et al., 2018; Efroni et al., 2021), inputs (Mao et al., 2018), noncontrollable states (Pan et al., 2022)). The works in reinforcement learning that deal with the presence of such dynamic contexts are referred to as situational RL (Chen et al., 2022) in this paper.

Existing works in situational RL have focused on the challenges of efficient learning and planning with the factorized dynamics induced by context transitions, e.g., variance reduction (Mao et al., 2018), separating contexts and endogenous states from observations (Dietterich et al., 2018; Chitnis & Lozano-Perez, 2019; Efroni et al., 2021; Pan et al., 2022) to learn more efficiently, detecting the abrupt changes of latent contexts (Chen et al., 2022). However, robustness against context disturbances is overlooked by existing works.

Robustness is critical for real-world tasks since RL policies are often brittle when faced with even slight variations in their environments (Meng & Khushi, 2019; Lu et al., 2020). We emphasize that we are considering *robustness against deviations of context transitions*, which is of particular interest in many real-world scenarios. For example, in Adaptive Cruise Control (ACC), an autonomous driving scenario, the speed of lead car can be viewed as context whose transition can be influenced by numerous factors but out of the control of the ego car. However, after making a decision, the state of the ego car is clear. Here a factorized structure is revealed, where deviations of context transitions are dominant while state transitions of the ego car contain almost no uncertainty. A robust and relatively conservative decision of the ego car is important to avoid crashing.

This is in contrast with existing works in robust RL, which tackle the discrepancy of the entire transitions between source and target domain, via various different approaches,

[*] This work is conducted at Microsoft. [1] Department of Mathematical Sciences, Tsinghua University [2] Rotman business school, University of Toronto [3] Microsoft Research Asia [4] Yau Mathematical Sciences Center and Department of Mathematical Sciences, Tsinghua University. Correspondence to: Li Zhao <lizo@microsoft.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

e.g., robust MDP (Roy et al., 2017; Wang & Zou, 2021), robust adversarial training (Pinto et al., 2017; Kamalaruban et al., 2020; Tessler et al., 2019), domain randomization (Andrychowicz et al., 2018; Peng et al., 2017), etc. See Section 5 for more detailed discussion. The robust RL approach might not give a policy that generalizes well enough across various context transitions since they do not take disturbances in context part into account in a precise way, and considering the deviations of the whole transition is too coarse to provide enough information about how to tackle the changes of context transitions.

We propose the framework of robust situational Markov decision process (RS-MDP) using Huber's contamination model (Huber, 1965) as the uncertainty set modeling the possible deviations of context transitions. This means that, with small probability, the context transition will change to an arbitrary distribution over the context space. We have to consider the worst-case future context at each step during the Bellman backup and thus minimization over all possible contexts is required to learn a robust Q-function. The challenge here is that the continuous and high-dimensional contexts make it hard to calculate the minimization operator over context space. To tackle this problem, we introduce the softmin smoothed robust Bellman operator, which leverages the factorized structure in system dynamics and does not disturb the endogenous transition, to approximate the robust Q-function. We prove an upper bound on the approximation error to validate our approach theoretically. Further we extend Soft Actor-Critic (SAC, Haarnoja et al., 2018) to robust situational SAC (RS-SAC), by modifying the policy evaluation step with a robust situational update, to learn robust situational policies. Our method enjoys one additional benefit that it is simple to implement and does not require task-specific prior knowledge on environment parameters or specially designed simulators to model the disturbance as done in adversarial training (Pinto et al., 2017) or domain randomization (Tobin et al., 2017). Finally we evaluate our algorithm on MuJoCo tasks (Todorov et al., 2012) with dynamic contexts and inventory control tasks.

The contributions of this paper are summarized as follows:

- (Section 2) We introduce robust situational MDP which captures the disturbances in context transitions explicitly and is suitable for many real-world applications.
- (Section 3) To learn robust policies for situational RL with large context space, we introduce the softmin smoothed robust Bellman operator to approximate the robust Q-value, and extend SAC to RS-SAC with a robust situational update in the policy evaluation step.
- (Section 4) Experiments on MuJoCo tasks with dynamic contexts and inventory control tasks show that our algorithm can generalize better to various context transitions and outperform existing robust RL algorithms.

2. Problem Formulations

In this section, we define robust situational MDP (RS-MDP) as a tuple $(\mathcal{S}, \mathcal{Z}, \mathcal{A}, M, \mathcal{U}, r, \gamma, \rho)$ where $\mathcal{S}$ is the (endogenous) state space, $\mathcal{Z}$ is the context space, $\mathcal{A}$ is the action space, $M : \mathcal{S} \times \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{S} \times \mathcal{Z})$ [1] is the transition kernel, $\mathcal{U}$ is the uncertainty set containing possible deviations of context transition, $r : \mathcal{S} \times \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$ is the reward function and $\gamma \in (0, 1)$ is the discount factor. Following the setting of situational RL (Dietterich et al., 2018; Mao et al., 2018; Pan et al., 2022), the transition kernel can be factorized as

$$M(s', z'|s, z, a) = \bar{P}(z'|z)P(s'|s, z, a), \tag{1}$$

for any $s \in \mathcal{S}, z \in \mathcal{Z}, a \in \mathcal{A}$, where $\bar{P} : \mathcal{Z} \to \Delta(\mathcal{Z})$ is the context transition, $P : \mathcal{S} \times \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the endogenous transition.

To capture the possible contamination to context transitions under real-world scenarios, we impose the Huber's contamination model (Huber, 1965) on the context transition $\bar{P}$ in (1) and define the uncertainty set to be

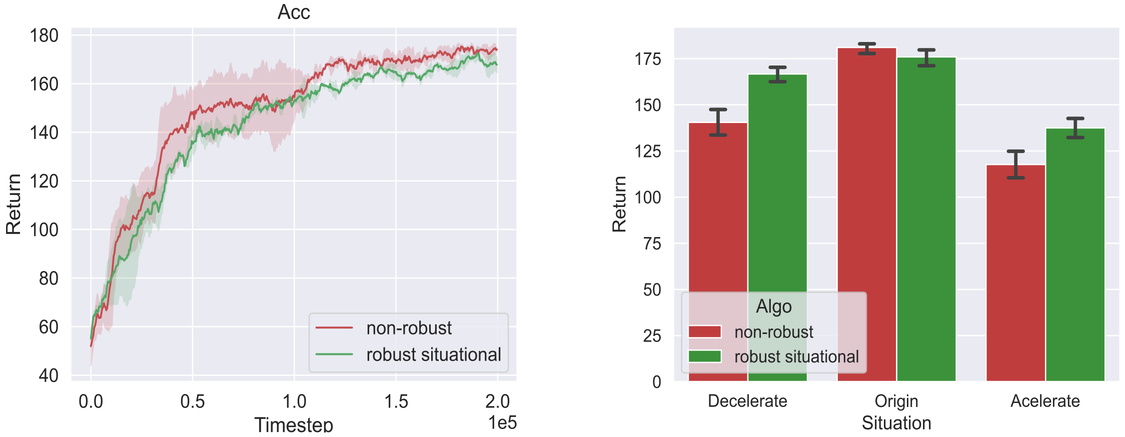
$$\mathcal{U}_{s,z,a} = \Big\{ \big((1 - \beta)\bar{P}(z'|z) + \beta q(z')\big) P(s'|s, z, a) \\ |q(\cdot) \in \Delta(\mathcal{Z}) \Big\}, \tag{2}$$

for any s, z, a and $\beta \in [0, 1)$. The definition of the uncertainty set $\mathcal{U}_{s,z,a}$ indicates that, with probability β, the context transition will change to an arbitrary distribution over the context space $\mathcal{Z}$. Intuitively, there is a context player selecting arbitrary context transitions from the uncertainty set $\mathcal{U}_{s,z,a}$ to disturb the agent. Note that the q chosen by the context player can depend on s, z and a. This kind of uncertainty set model is widely used in the literature of robust statistics and optimization (Huber, 1965; Du et al., 2018; Prasad et al., 2020). The goal of RS-MDP is to learn a single policy that maximizes the worst-case expected return with respect to possible context transitions in the uncertainty set defined in Equation (2). Therefore, given a policy π, we define the robust Bellman operator following (Iyengar, 2005) as

$$\mathcal{B}^{\pi}_{\text{rob}} Q(s, z, a) := r(s, z, a) + \\ \gamma \min_{\widetilde{M} \in \mathcal{U}_{s,z,a}} \mathbb{E}_{s',z',a'}[Q(s', z', a')], \tag{3}$$

where $s', z' \sim \widetilde{M}(\cdot|s, z, a)$ and $a' \sim \pi(\cdot|s', z')$ under the expectation. It is a γ-contraction whose fixed point Q^{π}_{rob}

[1] $\Delta(B)$ denotes the set of probability distributions over set B.



(a) Training curves

(b) Performance in three scenarios

Figure 1. Performance comparison of robust situational Q-learning and non-robust Q-learning in the tabular ACC environment.

is the robust Q-function (see e.g., Iyengar, 2005; Nilim & Ghaoui, 2003). Specifically, Equation (3) can be written as

$$\begin{aligned}\mathcal{B}^{\pi}_{\text{rob}}Q(s,z,a) &= r(s,z,a) + \gamma(1-\beta)\mathbb{E}_{s',z',a'}[Q(s',z',a')] \\ &+\gamma\beta \min_{q\in\Delta(\mathcal{Z})}\int_{\mathcal{Z}}\mathbb{E}_{s',a'}[Q(s',z'',a')]q(z'')dz'' \\ &= r(s,z,a) + \gamma(1-\beta)\mathbb{E}_{s',z',a'}[Q(s',z',a')] \\ &+\gamma\beta \min_{z''}\mathbb{E}_{s',a'}[Q(s',z'',a')]\end{aligned} \tag{4}$$

where $s', z' \sim M(\cdot|s,z,a)$ now. The minimization in Equation (4) implies that selecting the worst-case context transition from the uncertainty set defined in Equation (2) by the context player is equivalent to giving rise to a worst-case future context and the magnitude of this disturbance is limited by the constant β. Note that when $\beta = 0$, we recover the usual Bellman operator.

Our proposed uncertainty set in Equation (2) precisely captures the setting where only deviations of the context transitions matter, which is well motivated by real-world applications. For example, in inventory control, the customer demand is the context that suffers high randomness. It is possible to merge states and contexts together to reduce to a standard robust MDP problem handled by common robust RL algorithms. However considering the worst-case on the state transitions (or transitions in a composite state space) will result in an overly conservative value estimation and thus hurt the performance. For example, in ACC, the worst-case state transition should be crashing in the next time step, which makes the value significantly underestimated and does not give information about possible changes of context transition. Based on this intuition, we show that how our approach improves the worst-case bound theoretically. See Appendix A.2 for detailed discussion.

2.1. A Motivating Example

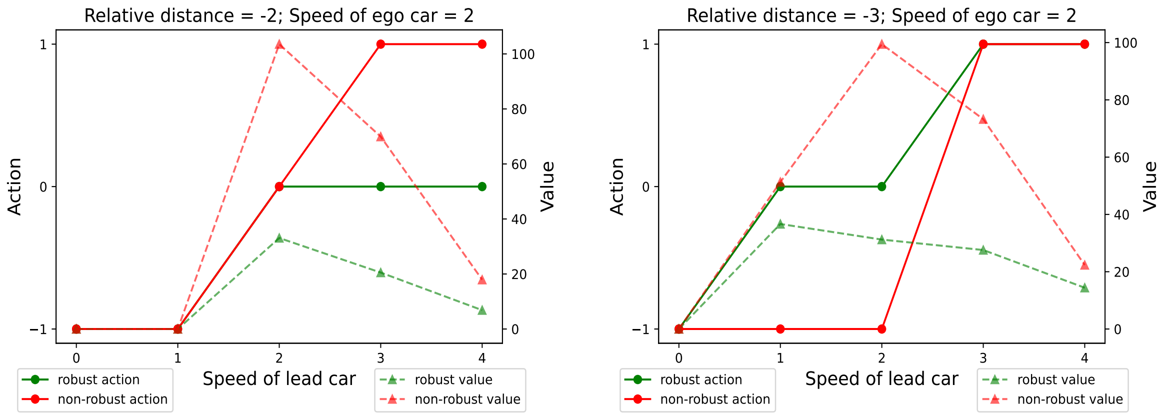
In this section, we design a simple tabular ACC environment to show that our proposed framework RS-MDP can indeed handle the changes of context transitions. The goal of ACC is to follow a lead car as closely as possible without crashing into it. The context space $\mathcal{Z}$ is the speed of lead car $z \in [0,5]$. The state $s = (v_e, d) \in \mathcal{S}$ consists of the speed of ego car $v_e \in [0,5]$ and the relative distance $d \in [-10, 0]$. All variables are integers. There are three actions: -1 (deceleration), 0 (doing nothing), 1 (acceleration). The reward function $r(s,z,a) = 10 + d$. When $d = -10$ (staying too far) or $d = 0$ (crashing), the episode is terminated. The maximum length of each episode is 20. At each time step the lead car samples an acceleration rate $\Delta z \in \{-1, 0, 1\}$ from a distribution $p = (p_{-1}, p_0, p_1)$ with $\Pr[\Delta z = i] = p_i$ for $i = -1, 0, 1$.

Considering the worst-case one-step future context taken in the Bellman backup in our RS-MDP framework, we propose the robust situational Q-learning algorithm to solve the ACC task

$$\begin{aligned}&Q_{t+1}(s_t,z_t,a_t) \leftarrow (1-\alpha_t)Q_t(s_t,z_t,a_t) \\ &\ +\alpha_t(r_t + \gamma(1-\beta)V_t(s_{t+1},z_{t+1}) + \gamma\beta\cdot\min_{z} V_t(s_{t+1},z))\end{aligned} \tag{5}$$

where $V_t(s,z) = \max_a Q_t(s,z,a)$ and each action $a_t \sim \pi_b(\cdot|s_t,z_t)$ is sampled from a behavior policy π_b. Again when $\beta = 0$, Equation (5) reduces to the non-robust Q-learning.

We run robust situational Q-learning and non-robust Q-learning on the ACC environment for 200k steps separately. We set $\gamma = 0.99$, $\alpha_t = 0.1$, $\beta = 0.2$ and π_b a uniformly random policy in the experiments. The context dynamics of the



(a) Relative distance = -2

(b) Relative distance = -3

Figure 2. The behaviors for robust situational Q-learning and non-robust Q-learning in the tabular ACC environment. The left y-axis is the chosen actions and the right y-axis is the corresponding Q-values.

nominal training environment is set to be $p = (0.4, 0.2, 0.4)$. We show the training curve in Figure 1(a), which implies that, though robust situational Q-learning aims at optimizing the performance on worst-case contexts, it has only slight impact on the performance in the nominal environment compared with the non-robust one. For robustness analysis, we further perturb the context transition to result in the Decelerate scenario with $p = (0.7, 0, 0.3)$ and the Accelerate scenario with $p = (0.3, 0, 0.7)$, implying that the lead car is more likely to decelerate and accelerate, respectively. As shown in Figure 1(b), robust situational Q-learning outperforms the non-robust one in these new scenarios, since our algorithm aims at optimizing the worst-case situations.

To gain more insights on the results, we fix the state $s = (d, v_e)$, relative distance and speed of ego car, and then check the actions and corresponding Q-values given by two algorithms for different contexts, shown in Figure 2, to study the behaviors of policies. Observe that the Q-values given by the robust situational Q-learning are significantly lower than the non-robust one, which is due to the robust Bellman backup by considering worst-case future contexts in Equation 5. This indeed induces the desired conservative behavior: If relative distance is -2 (cf. Figure 2(a)), the robust action do not take action 1 to accelerate ego car like the non-robust one even when the speed of lead car is high. Two cars now are so close that it would be better to not accelerate, in order to avoid crashing incurred by possibly sudden deceleration of the lead car. But if relative distance is -3 (cf. Figure 2(b)), two cars are in a safer distance and now the robust action does not show conservativeness as in previous case.

3. Deep Robust Situational RL

It is common in practice that the context space $\mathcal{Z}$ is large or even continuous, and so the min operator over $\mathcal{Z}$ in Equation (4) is in general intractable. To overcome this difficulty, we modify the original robust Bellman operator defined in Equation (4) and introduce softmin smoothed robust Bellman operators in Section 3.1 to approximately solve the minimization. Then, we apply our RS-MDP framework to existing deep RL algorithm to learn robust situational policies in Section 3.2.

3.1. Smoothed by Softmin Operators

For convenient, we define the *softmin* operator, for any function $f : \mathcal{Z} \to \mathbb{R}$,

$$\begin{aligned}&\mathrm{SoftMin}_z(f(z))\\&:= \int_{\mathcal{Z}} f(z)\exp(-\frac{1}{\tau}f(z))dz \Big/ \int_{\mathcal{Z}} \exp(-\frac{1}{\tau}f(z'))dz'\end{aligned} \tag{6}$$

where $\tau > 0$ and as $\tau \to 0$, the softmin operator is approaching the min operator. We introduce the softmin smoothed robust Bellman operator

$$\begin{aligned}\mathcal{B}_\tau^\pi Q(s,z,a) = r(s,z,a) + \gamma(1-\beta)\mathbb{E}_{s',z',a'}[Q(s',z',a')]\\ + \gamma\beta \cdot \mathrm{SoftMin}_{z'}\Big(\mathbb{E}_{s',a'}[Q(s',z',a')]\Big)\end{aligned} \tag{7}$$

Note that $\mathcal{B}_\tau^\pi$ is not necessarily a γ-contraction since the softmin operator is not guaranteed to be non-expansive (Littman, 1996). Instead, we obtain an error bound between the true robust value Q^π_{rob} and the t-th iteration $Q_t := \mathcal{B}_\tau^\pi Q_{t-1}$ starting from Q_0:

Theorem 3.1. *For any function $f : \mathcal{Z} \to \mathbb{R}$, let $\mathcal{C}(f, \epsilon) :=$*

$\{z \in \mathcal{Z} \mid f(z) \leq \min_z f(z) + \epsilon\}$, where $\epsilon > 0$. Let $Q_t = \mathcal{B}^{\pi}_{\tau} Q_{t-1}$ to be the t-th iteration and fix $\epsilon > 0$. Then the difference between Q_t and the optimal robust Q-function Q^{π}_{rob} satisfies

$$||Q_t - Q^{\pi}_{\text{rob}}||_{\infty} \leq \gamma^t ||Q_0 - Q^{\pi}_{\text{rob}}||_{\infty} + \frac{\beta}{1-\gamma}(C_{\max}(\epsilon) \cdot \tau + \epsilon) \tag{8}$$

where $C_{\max}(\epsilon)$ is given by

$$C_{\max}(\epsilon) := \max_{s,z,a,k} \Big(\int_{\mathcal{Z}} 1 dz - 1 - \log \int_{\mathcal{C}(F^k_{s,z,a}, \epsilon)} 1 dz \Big) \tag{9}$$

and $F^k_{s,z,a}(z') := \mathbb{E}_{s' \sim P(\cdot|s,z,a), a' \sim \pi(\cdot|s',z')}[Q_k(s', z', a')]$.

The proof can be found in Appendix A.1. By fixing $\epsilon > 0$, the error between the value function induced by the softmin operator and the true robust value will converge to $\beta\epsilon/(1-\gamma)$ as $\tau \to 0$ and then can be arbitrarily close to 0 by taking ϵ small enough. Thus, in the policy evaluation step, the softmin operator gives a reasonable approximation and will benefit the robust training. However, the softmin operator involves integral which is intractable over continuous context space $\mathcal{Z}$. We use importance sampling (Haarnoja et al., 2017) in expectation to rewrite the integral and obtain an unbiased estimation. Specifically,

$$\begin{aligned} &\text{SoftMin}_{z'}(f(z')) \\ &= \mathbb{E}_{z' \sim q}\Big[\frac{f(z') \exp(-\frac{1}{\tau} f(z'))}{q(z')}\Big] / \mathbb{E}_{z' \sim q}\Big[\frac{\exp(-\frac{1}{\tau} f(z'))}{q(z')}\Big] \end{aligned} \tag{10}$$

where $q \in \Delta(\mathcal{Z})$ is a sampling distribution over the context space $\mathcal{Z}$ and $f(z') = \mathbb{E}_{s',a'}[Q(s', z', a')]$. In practice, we sample contexts z' by adding noises, which are sampled from the uniform distribution $\epsilon \sim \text{Unif}(-c, c)$ in the range $[-c, c]$, to the true context z'_{env} obtained from the environment, i.e., $z' = z'_{\text{env}} + \epsilon$. We will call c the noise clip parameter, which represents the possible maximal deviation of the context and is meaningful in real world, e.g., the outside air temperature will not change too dramatically in short time and will fall in some reasonable range.

3.2. Robust Situational SAC: An Instance

To verify the utility of the RS-MDP framework, we apply it to the existing RL algorithm, Soft Actor-Critic (SAC, Haarnoja et al., 2018), and obtain an RS-MDP based algorithm called Robust Situational Soft Actor-Critic (RS-SAC) to learn robust policies against context disturbances. The general idea is to modify the policy evaluation step to be the robust situational one, i.e., letting the critic network approximate the robust Q-value following the softmin smoothed Bellman operator defined in Equation (7) and keeping the policy improvement step unchanged. Note that in a similar way our RS-MDP framework can be combined with a wide range of base RL algorithms which involve learning a Q-function.

The original SAC is an off-policy actor-critic algorithm based on maximum entropy principle. Let $\phi_j, j = 1, 2$, and θ be the parameters of the Q-networks and the policy network, respectively. Following the implementation in (Haarnoja et al., 2018), the policy is reparametrized as $a_t = f_{\theta}(s_t, z_t; \xi)$, where $\xi \sim \mathcal{N}$ is standard Gaussian noise, and the target Q-network Q_{ϕ^-} is applied with soft update. We define $Q^{\min}_{\phi} := \min_{j=1,2} Q_{\phi_j}$ and similarly $Q^{\min}_{\phi^-} := \min_{j=1,2} Q_{\phi^-_j}$.

In the policy evaluation step, the goal is to learn the worst-case robust value function Q^{π}_{rob} (Mankowitz et al., 2020) under the RS-MDP framework. We perform the critic update by minimizing

$$\min_{\phi_j} \mathbb{E}_{\substack{s_t, z_t, a_t, r_t, s_{t+1}, z_{t+1} \sim \mathcal{D} \\ \xi \sim \mathcal{N}}} \Big[\big(Q_{\phi_j}(s_t, z_t, a_t) - Q_{\text{targ}} \big)^2 \Big], \tag{11}$$

for $j = 1, 2$, where the target value is defined as

$$\begin{aligned} Q_{\text{targ}} = r_t + \gamma \Big[&(1-\beta) Q^{\min}_{\phi^-}\big(s_{t+1}, z_{t+1}, f(s_{t+1}, z_{t+1}; \xi)\big) \\ &+ \beta \cdot \text{SoftMin}_z \big(Q^{\min}_{\phi^-}\big(s_{t+1}, z, f(s_{t+1}, z; \xi)\big)\big) \\ &- \alpha \log \pi_{\theta}\big(f_{\theta}(s_{t+1}, z_{t+1}; \xi) \mid s_{t+1}, z_{t+1}\big) \Big] \end{aligned} \tag{12}$$

where the SoftMin over the context space $\mathcal{Z}$ is performed by using importance sampling as in Equation (10). Note that we recover the original critic update of SAC by letting $\beta = 0$.

For the policy improvement step, we train the actor by maximizing

$$\begin{aligned} \max_{\theta} \mathbb{E}_{x_t \sim \mathcal{D}, \xi \sim \mathcal{N}} \Big[&Q^{\min}_{\phi}\big(x_t, f(x_t; \xi)\big) \\ &- \alpha \log \pi_{\theta}(f_{\theta}(x_t; \xi) \mid x_t) \Big], \end{aligned} \tag{13}$$

where $x_t = (s_t, z_t)$ is the total state. At last, we update the temperature α by minimizing

$$\min_{\alpha} -\alpha \mathbb{E}_{x_t \sim \mathcal{D}, \xi \sim \mathcal{N}}[\log \pi_{\theta}(f(x_t; \xi) \mid x_t) + H], \tag{14}$$

where H is the target value of entropy.

4. Experiments

In this section, we first conduct experiments on MuJoCo (Todorov et al., 2012) tasks with dynamic contexts in Section 4.1. Then, we apply our algorithm RS-SAC to the real-world inventory control task to show the wide applicability of our RS-MDP framework in Section 4.2.

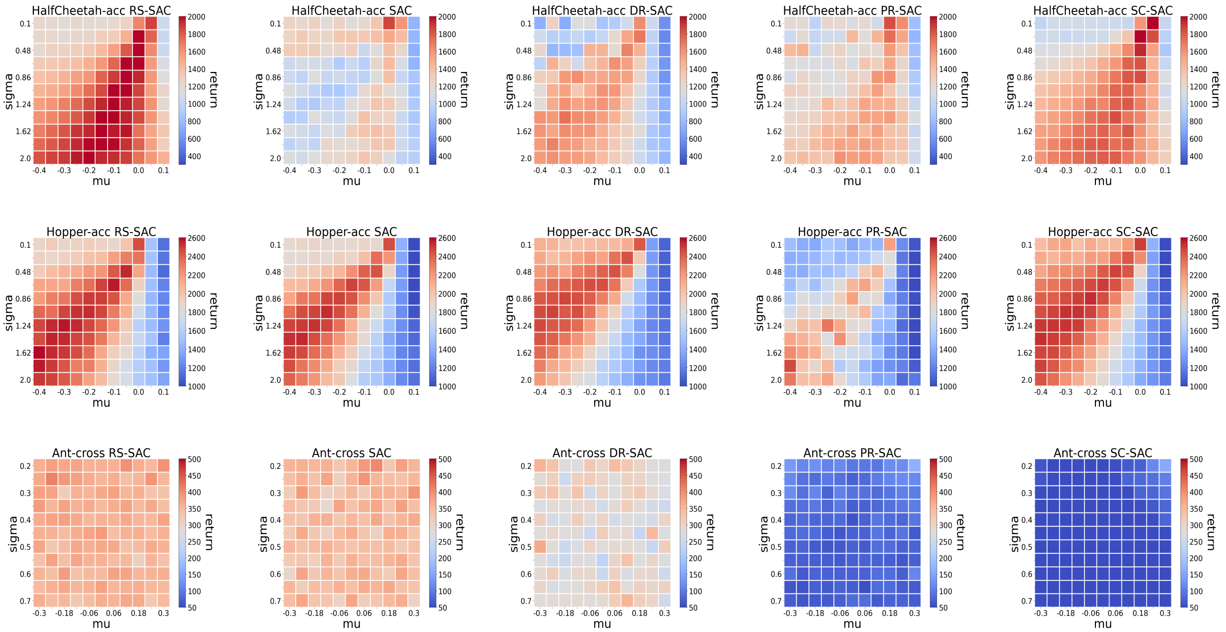


Figure 3. Compare RS-SAC, SAC, DR-SAC, PR-SAC and SC-SAC in the target context transitions with perturbed parameters μ and σ. Each point is the average return obtained by running 8 episodes for each policy obtained from the last 10 epochs in training for each random seed. We also fix several values for μ and plot the average return curve for each algorithm as a function of σ, see Appendix B.4. The results show that RS-SAC gives more robust policies across various context transitions than other baselines.

4.1. Locomotion Control Tasks with Dynamic Contexts

This section provides empirical study on several challenging locomotion control tasks with dynamic contexts to show that RS-SAC is able to handle complex environments. We modify standard MuJoCo (Todorov et al., 2012) tasks, which are commonly used in previous papers (Tessler et al., 2019; Kuang et al., 2022), by adding appropriate context spaces to make the assumption of situational RL (cf. Equation 1) holds.

Motivated by the ACC example, we design two environments, HalfCheetah-acc and Hopper-acc. We assume there is a lead car whose speed v_t is the context and require the agent to stay close to the lead car and avoid crashing. The context transition is of the form $v_{t+1} = \bar{P}_{\Delta v}(v_t) = \max\{v_t + \Delta v, 0\}$, where $\Delta v \sim \mathcal{N}(\mu, \sigma)$ is the change of speed and we do not allow the lead car to go backward. Another environment, Ant-cross, assume there is a moving obstacle of radius r, whose x-position is fixed and y-position y_t is the context, between the agent and the goal position. The context transition here is i.i.d. $y_t \sim \mathcal{N}(\mu, \sigma)$. The policy needs to move across the obstacle without hitting to it and reach the goal position as soon as possible. In these environments, once the agent gets contact with the lead car or the obstacle, the environment will terminate the episode and return a penalty. See Appendix B.1 for more details about the environments.

For baselines, we use SC-SAC (Kuang et al., 2022), which optimizes over the worst-case disturbance in state-context space, as a representative robust RL algorithm. We also choose PR-SAC (Tessler et al., 2019), which uses an adversarial policy to disturb actions during training, to represent robust RL algorithm with adversarial training. The reason for selecting PR-SAC is that, for example, the unexpected deceleration of lead car can be regarded as that the action of the agent is perturbed by the adversary to give an acceleration. We adapt domain randomization (DR) to SAC, called DR-SAC, which randomizes parameters μ and σ in source environments during training to learn policies that are robust to perturbations on these parameters. We train policies for 1000k steps in all tasks with fixed hyperparameters. The parameters of nominal training environments are set to be $(\mu, \sigma) = (0, 0.2)$ for HalfCheetah-acc and Hopper-acc, and $(\mu, \sigma) = (0, 0.3)$ for Ant-cross. For testing, we perturb the parameters μ and σ in target environments to generate disturbances in context transitions. See Appendix B.2 for more details on implementation and evaluation settings, and see Appendix B.3 for the training curves about the algorithms.

We analyze robustness against deviations of context transitions for all algorithms. In Figure 3 we compare the performances of well-trained policies obtained from different algorithms on the target context transitions with perturbed parameters μ and σ, and in Figure 4, we summarize the

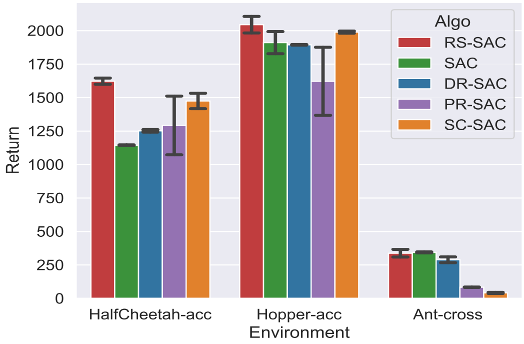


Figure 4. Average returns in target context transitions for all algorithms in our MuJoCo tasks. The error bars are over random seeds.

obtained heatmaps by plotting the averages over them. The results show that RS-SAC outperforms other algorithms in the target context transitions. For SAC, it is non-robust compared with RS-SAC except in Ant-cross. While DR-SAC shows more robustness than SAC since it has seen diverse context transitions during training, its generalization performance is not as good as our RS-SAC which is more principled. For PR-SAC, the results imply that generating disturbances in action space is not guaranteed to learn robust policies against context disturbances, though PR-SAC is more robust than SAC in HalfCheetah-acc. Compared with SC-SAC, our algorithm RS-SAC shows competitive robustness in Hopper-acc and outperforms SC-SAC in other two tasks, since SC-SAC does not utilize the fact that there is no need to optimize over worst-case disturbances in state space.

We provide ablation study on the noise clip parameter c used in Equation (10), since c is an important parameter, which determines our searching range for the worst-case context, centered around the true context obtained from the environment. Though we theoretically establish the RS-MDP framework by considering the worst-case context from the whole context space in Equation (2), we find that in our experiments such a search over the whole context space might be harmful even if the context space is bounded. More detailed results can be found in Appendix B.5.

4.2. Application to Inventory Control Tasks

In this section, we apply RS-SAC to the inventory control task, where customer demand is the context variable, to show the wide applicability of our RS-MDP framework.

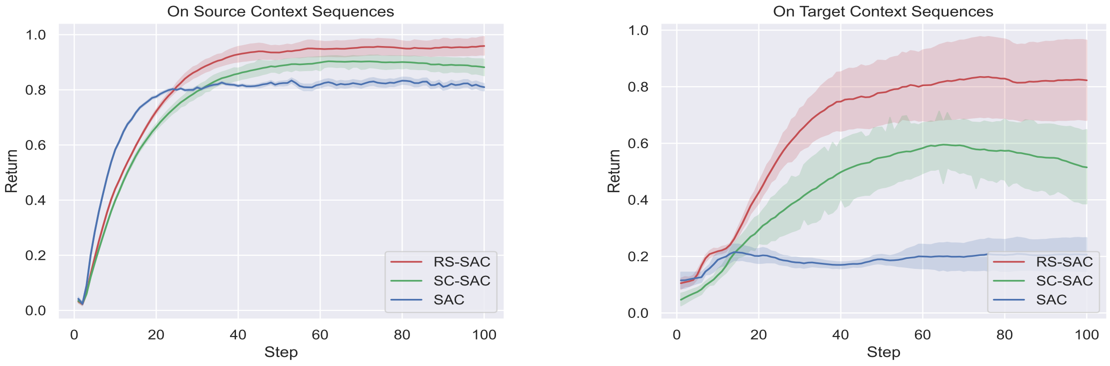
One common practice to apply deep RL to inventory control is to build simulator with historical data of customer demands to train an RL policy (Gijsbrechts et al., 2022). However, during deployment, the context transitions could be influenced by numerous factors, e.g., seasons, trends, etc., and thus deviate from the seen context sequences in training. Therefore robustness against deviations of context transitions should improve the performance for RL in inventory control.

To establish experiments, historical data of customer demands from 50 Stock Keeping Units (SKUs) are used to build the training simulators, and fixed sequences of customer demands from other 5 SKUs serve as target domains to test RL policies. The action is the number of placing orders, which will suffer a time lag, called lead time, between the actual placement of the order and the arrival of the items in the warehouse. We will fix the lead time in experiments. The state-context space consists of the incoming orders of dimension equal to lead time, the in-stock levels and historical demands of length 3. The goal is to maximize the profit and minimize the cost originated from restocking, backlogs, etc. We compare RS-SAC with original SAC and SC-SAC (Kuang et al., 2022). All algorithms are trained for 400k steps. For more details on the environment and implementation, see Appendix C.

Figure 5 shows that RS-SAC outperforms other algorithms, which indicates the benefits of robust situational training. We show the behaviors of different algorithms in one target context sequence of length 100 in Figure 6. There are spikes in the context sequence and the demands are occasionally near 500, which are uncommon. RS-SAC gradually increases its placing orders to satisfy the suddenly increased demand, since it considers the worst-case future context which will be high demands. However, SAC simply places rather less orders to reduce restocking cost, which is suboptimal, and SC-SAC fails to adapt to this unseen sequence of customer demands and could not respond to the occasionally high demands in time.

5. Related Works

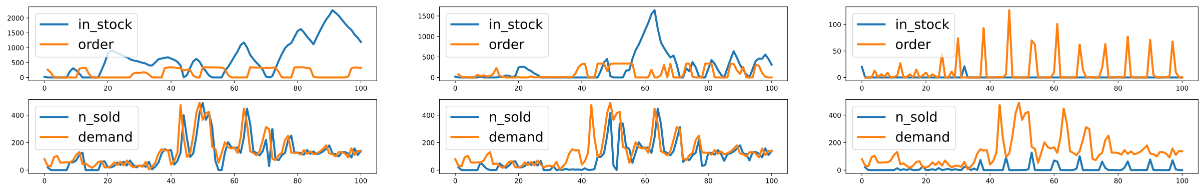
Situational RL: Existing works on situational RL focus on utilizing the factorized dynamics induced by context transitions to achieve efficient learning and planning (Chitnis & Lozano-Perez, 2019; Efroni et al., 2021). Specifically, Mao et al. (2018) proposes to use input-dependent baselines, which is a function of both the state and the entire future input sequence, to reduce variance in policy gradient methods. Dietterich et al. (2018) argues that separating contexts and endogenous states from observations can benefit the learning process in certain cases. Chen et al. (2022) proposes SeCBAD, a Bayes adaptive RL approach, to detect the abrupt changes of latent contexts. Pan et al. (2022) improves the learning of world model by decoupling the controllable and uncontrollable state transitions and then benefits the decision-making of RL. However, none of these works consider the problem of robustness against context



(a) Performance on the first 50 SKUs

(b) Performance on the other 5 SKUs

Figure 5. Performance comparison on the inventory control task. The returns are normalized.



(a) RS-SAC

(b) SC-SAC

(c) SAC

Figure 6. Behavior study of different algorithms on the inventory control task.

disturbances like ours.

Robust RL: The theoretical framework of robust RL is based on robust MDPs which was introduced in (Jay K. Satia, 1973; Chelsea C. White, 1994; Nilim & Ghaoui, 2003; Iyengar, 2005). In practice, uncertainty sets can be represented by multiple simulators. Specifically, Peng et al. (2017); Tobin et al. (2017); Andrychowicz et al. (2018) propose domain randomization (DR) which randomly generates different transition dynamics to facilitate robust policy training. Mankowitz et al. (2020) proposes robust MPO (R-MPO) to learn robust policies with pre-given multiple environment parameters as uncertainty sets. Abdullah et al. (2019) proposes the Wasserstein robust RL (WR2L) to train the policy with environment parameters jointly. Jiang et al. (2021) propose monotonic robust policy optimization (MRPO) which optimizes a theoretical performance lower bound to learn robust policies with increasing performance. Different from the above approaches which require multiple simulators to model the disturbance in transition dynamics, Kuang et al. (2022) proposes state-conservative policy optimization (SCPO) which optimizes over the worst-case disturbance in state space and is free from specific priors and control of simulators.

Another branch of robust RL is robust adversarial training where an adversary is explicitly modeled to generate disturbances to the environments to learn robust policies. The first line of works relies on the access to the simulator where the transition dynamics can be modified by an adversary (Pinto et al., 2017; Zhang et al., 2020; Kamalaruban et al., 2020; Tessler et al., 2019; Zhai et al., 2022; Tanabe et al., 2022). Instead of modifying the simulator directly, another work (Tessler et al., 2019) proposes action robust MDP (AR-MDP) which models the disturbances in environments through perturbations on the action space.

6. Conclusions and Future Works

This paper introduces robust situational MDP which captures the deviations of context transitions explicitly and is suitable for many real-world applications. To scale to large context space, we introduce the softmin smoothed robust Bellman operator to learn the robust Q-value approximately, and experiments show that our algorithm can generalize better to a wide range of deviations in context transitions and outperform existing robust RL algorithms. For future work, we would like to apply our method to more real-world tasks, and delve into the specific structures revealed by applications to design effective robust algorithms.

7. Acknowledgements

JZ thanks all co-authors for their patience, guidence, valuable works and discussion. JZ also thanks Shihui Zhao for her encouragement.

References

Abdullah, M., Ren, H., Bou-Ammar, H., Milenkovic, V., Luo, R., Zhang, M., and Wang, J. Wasserstein robust reinforcement learning. *ArXiv*, abs/1907.13196, 2019.

Andrychowicz, M., Baker, B., Chociej, M., Józefowicz, R., McGrew, B., Pachocki, J. W., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., and Zaremba, W. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39:20 – 3, 2018.

Chelsea C. White, III, H. K. E. Markov decision processes with imprecise transition probabilities. *Operations Research*, 42(4):739–749, 1994.

Chen, X., Zhu, X., Zheng, Y., Zhang, P., Zhao, L., Cheng, W., CHENG, P., Xiong, Y., Qin, T., Chen, J., and Liu, T.-Y. An adaptive deep RL method for non-stationary environments with piecewise stable context. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=5swt6zUFrVp.

Chitnis, R. and Lozano-Perez, T. Learning compact models for planning with exogenous processes. In *CoRL*, 2019.

Dietterich, T. G., Trimponias, G., and Chen, Z. Discovering and removing exogenous state variables and rewards for reinforcement learning. *ArXiv*, abs/1806.01584, 2018.

Du, S. S., Wang, Y., Balakrishnan, S., Ravikumar, P., and Singh, A. Robust nonparametric regression under huber's ϵ-contamination model. *ArXiv*, abs/1805.10406, 2018.

Efroni, Y., Misra, D. K., Krishnamurthy, A., Agarwal, A., and Langford, J. Provable rl with exogenous distractors via multistep inverse dynamics. *ArXiv*, abs/2110.08847, 2021.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *ArXiv*, abs/1802.09477, 2018.

Gijsbrechts, J., Boute, R. N., Mieghem, J. A. V., and Zhang, D. J. Can deep reinforcement learning improve inventory management? performance on lost sales, dual-sourcing, and multi-echelon problems. *Manuf. Serv. Oper. Manag.*, 24:1349–1368, 2022.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, 2017.

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

Huber, P. J. A robust version of the probability ratio test. *Annals of Mathematical Statistics*, 36:1753–1758, 1965.

Iyengar, G. N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

Jay K. Satia, Roy E. Lave, J. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, 1973.

Jiang, Y., Li, C., Dai, W., Zou, J., and Xiong, H. Monotonic robust policy optimization with model discrepancy. In *ICML*, 2021.

Kamalaruban, P., Huang, Y.-T., Hsieh, Y.-P., Rolland, P., Shi, C., and Cevher, V. Robust reinforcement learning via adversarial training with langevin dynamics. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8127–8138, 2020.

Kuang, Y., Lu, M., Wang, J., Zhou, Q., Li, B., and Li, H. Learning robust policy against disturbance in transition dynamics via state-conservative policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Littman, M. L. Algorithms for sequential decision making. PhD thesis, Department of Computer Science, Brown University, 1996.

Lu, M., Shahn, Z., Sow, D. M., Doshi-Velez, F., and wei H. Lehman, L. Is deep reinforcement learning ready for practical applications in healthcare? a sensitivity analysis of duel-ddqn for hemodynamic management in sepsis patients. *AMIA 2020 Annual Symposium proceedings. AMIA Symposium*, pp. 773–782, 2020.

Mankowitz, D. J., Levine, N., Jeong, R., Abdolmaleki, A., Springenberg, J. T., Mann, T., Hester, T., and Riedmiller, M. A. Robust reinforcement learning for continuous control with model misspecification. *ArXiv*, abs/1906.07516, 2020.

Mao, H., Venkatakrishnan, S. B., Schwarzkopf, M., and Alizadeh, M. Variance reduction for reinforcement learning in input-driven environments. *ArXiv*, abs/1807.02264, 2018.

Meng, T. L. and Khushi, M. Reinforcement learning in financial markets. *Data*, 4:110, 2019.

Nilim, A. and Ghaoui, L. E. Robustness in markov decision problems with uncertain transition matrices. In *Advances in Neural Information Processing Systems 16 (NIPS'03)*. MIT Press, 2003.

Pan, M., Zhu, X., Wang, Y., and Yang, X. Iso-dream: Isolating noncontrollable visual dynamics in world models. volume abs/2205.13817, 2022.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2017.

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. K. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, 2017.

Prasad, A., Srinivasan, V., Balakrishnan, S., and Ravikumar, P. On learning ising models under huber's contamination model. In *Neural Information Processing Systems*, 2020.

Roy, A., Xu, H., and Pokutta, S. Reinforcement learning under model mismatch. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Tanabe, T., Sato, R., Fukuchi, K., Sakuma, J., and Akimoto, Y. Max-min off-policy actor-critic method focusing on worst-case robustness to model misspecification. *ArXiv*, abs/2211.03413, 2022.

Tessler, C., Efroni, Y., and Mannor, S. Action robust reinforcement learning and applications in continuous control. *ArXiv*, abs/1901.09184, 2019.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Wang, Y. and Zou, S. Online robust reinforcement learning with model uncertainty. In *Advances in Neural Information Processing Systems*, volume 34, pp. 7193–7206, 2021.

Zhai, P., Luo, J., Dong, Z., Zhang, L., Wang, S., and Yang, D. Robust adversarial reinforcement learning with dissipation inequation constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36(5), pp. 5431–5439, 2022.

Zhang, K., Hu, B., and Başar, T. On the stability and convergence of robust adversarial reinforcement learning: A case study on linear quadratic systems. In *NeurIPS*, 2020.

A. Proofs

A.1. Proof of Theorem 3.1

For ease of notations, we will use SM_τ to represent the softmin operator here. Let $\mathrm{LSE}_\tau(f(\cdot)) := -\tau \log \int_{\mathcal{Z}} \exp\Big(-\frac{1}{\tau} f(z)\Big) dz$ be the logsumexp operator.

Lemma A.1. *For any function $f : \mathcal{Z} \to \mathbb{R}$, let $\mathcal{C}(f, \epsilon) := \{z \in \mathcal{Z} \mid f(z) \leq \min_z f(z) + \epsilon\}$, where $\epsilon > 0$. Then*

$$\min_z f(z) - \tau \log \int_{\mathcal{Z}} 1dz \leq \mathrm{LSE}_\tau(f(\cdot)) \leq \min_z f(z) + \epsilon - \tau \log \int_{\mathcal{C}(f,\epsilon)} 1dz \tag{15}$$

Proof.

$$\begin{aligned}
\mathrm{LSE}_\tau(f(\cdot)) &= -\tau \log \int_{\mathcal{Z}} \exp\Big(-\frac{1}{\tau} f(z)\Big) dz \\
&\leq -\tau \log \int_{\mathcal{C}(f,\epsilon)} \exp\Big(-\frac{1}{\tau} f(z)\Big) dz \\
&\leq -\tau \log \int_{\mathcal{C}(f,\epsilon)} \exp\Big(-\frac{1}{\tau}(\min_{z'} f(z') + \epsilon)\Big) dz \\
&\leq \min_z f(z) + \epsilon - \tau \log \int_{\mathcal{C}(f,\epsilon)} 1dz
\end{aligned} \tag{16}$$

On the other hand,

$$\begin{aligned}
\mathrm{LSE}_\tau(f(\cdot)) &= -\tau \log \int_{\mathcal{Z}} \exp\Big(-\frac{1}{\tau} f(z)\Big) dz \\
&\geq -\tau \log \int_{\mathcal{Z}} \exp\Big(-\frac{1}{\tau} \min_{z'} f(z')\Big) dz \\
&= \min_z f(z) - \tau \log \int_{\mathcal{Z}} 1dz
\end{aligned} \tag{17}$$

□

Lemma A.2. *For any function $f : \mathcal{Z} \to \mathbb{R}$, let $\mathcal{C}(f, \epsilon) := \{z \in \mathcal{Z} \mid f(z) \leq \min_z f(z) + \epsilon\}$, where $\epsilon > 0$. Then*

$$0 \leq \mathrm{SM}_\tau(f(\cdot)) - \min_z f(z) \leq \tau(\int_{\mathcal{Z}} 1dz - 1 - \log \int_{\mathcal{C}(f,\epsilon)} 1dz) + \epsilon \tag{18}$$

Proof. The left-hand-side is straightforward. For the right-hand-side, let $p_\tau(z)$ be the distribution $p_\tau(z) := \frac{\exp(-\frac{1}{\tau} f(z))}{\int_{\mathcal{Z}} \exp(-\frac{1}{\tau} f(z)) dz}$. Then

$$\begin{aligned}
&\mathrm{SM}_\tau(f(\cdot)) - \mathrm{LSE}_\tau(f(\cdot)) \\
=& \int_{\mathcal{Z}} p_\tau(z) f(z) dz + \tau \log \int_{\mathcal{Z}} \exp\Big(-\frac{1}{\tau} f(z)\Big) dz \\
=& \tau\Big(\log \int_{\mathcal{Z}} \exp\Big(-\frac{1}{\tau} f(z)\Big) dz - \int_{\mathcal{Z}} p_\tau(z)(-\frac{1}{\tau} f(z)) dz\Big) \\
=& \tau \int_{\mathcal{Z}} -p_\tau(z) \log p_\tau(z) dz \\
\leq& \tau \int_{\mathcal{Z}} (1 - p_\tau(z)) dz \\
=& \tau(\int_{\mathcal{Z}} 1dz - 1)
\end{aligned} \tag{19}$$

where the inequality follows from the fact that $-x\log x \leq 1 - x$ for $0 < x \leq 1$. Finally we obtain that

$$\begin{aligned}
&\mathrm{SM}_\tau(f(\cdot)) - \min_z f(z) \\
=&\mathrm{SM}_\tau(f(\cdot)) - \mathrm{LSE}_\tau(f(\cdot)) + \mathrm{LSE}_\tau(f(\cdot)) - \min_z f(z) \\
\leq&\tau(\int_{\mathcal{Z}} 1dz - 1 - \log\int_{\mathcal{C}(f,\epsilon)} 1dz) + \epsilon
\end{aligned} \tag{20}$$

where the last inequality follows from equation (19) and Lemma A.1.

□

Proof of Theorem 3.1. For any $s \in \mathcal{S}$, $z \in \mathcal{Z}$, $a \in \mathcal{A}$,

$$\begin{aligned}
&\Big|Q_{t+1}(s,z,a) - Q^\pi_{\mathrm{rob}}(s,z,a)\Big| \\
=&\Big|\gamma(1-\beta)\mathbb{E}_{s',z',a'}[Q_t(s',z',a')] + \gamma\beta\cdot\mathrm{SM}_\tau(\mathbb{E}_{s',a'}[Q_t(s',\cdot,a')]) \\
&-\gamma(1-\beta)\mathbb{E}_{s',z',a'}[Q^\pi_{\mathrm{rob}}(s',z',a')] - \gamma\beta\cdot\min_{z'}\mathbb{E}_{s',a'}[Q^\pi_{\mathrm{rob}}(s',z',a')]\Big| \\
\leq&\gamma(1-\beta)\Big|\mathbb{E}_{s',z',a'}[Q_t(s',z',a')] - \mathbb{E}_{s',z',a'}[Q^\pi_{\mathrm{rob}}(s',z',a')]\Big| \\
&+\gamma\beta\cdot\Big|\mathrm{SM}_\tau(\mathbb{E}_{s',a'}[Q_t(s',\cdot,a')]) - \min_{z'}\mathbb{E}_{s',a'}[Q^*_{\mathrm{rob}}(s',z',a')]\Big|
\end{aligned} \tag{21}$$

Further, for the second term in the last inequality,

$$\begin{aligned}
&\Big|\mathrm{SM}_\tau(\mathbb{E}_{s',a'}[Q_t(s',\cdot,a')]) - \min_{z'}\mathbb{E}_{s',a'}[Q^\pi_{\mathrm{rob}}(s',z',a')]\Big| \\
\leq&\Big|\mathrm{SM}_\tau(\mathbb{E}_{s',a'}[Q_t(s',\cdot,a')]) - \min_{z'}\mathbb{E}_{s',a'}[Q_t(s',z',a')]\Big| \\
&+\Big|\min_{z'}\mathbb{E}_{s',a'}[Q_t(s',z',a')] - \min_{z'}\mathbb{E}_{s',a'}[Q^\pi_{\mathrm{rob}}(s',z',a')]\Big|
\end{aligned} \tag{22}$$

Define the norm $||Q||_\infty := \max_{s,z,a}|Q(s,z,a)|$ for function $Q : \mathcal{S}\times\mathcal{Z}\times\mathcal{A}\to\mathbb{R}$. Combining (21) with (22), we obtain

$$\begin{aligned}
&||Q_{t+1} - Q^\pi_{\mathrm{rob}}||_\infty \\
\leq&\gamma||Q_t - Q^\pi_{\mathrm{rob}}||_\infty + \gamma\beta\max_{s,z,a}\Big|\mathrm{SM}_\tau(\mathbb{E}_{s',a'}[Q_t(s',\cdot,a')]) - \min_{z'}\mathbb{E}_{s',a'}[Q^\pi_{\mathrm{rob}}(s',z',a')]\Big| \\
\leq&\gamma||Q_t - Q^\pi_{\mathrm{rob}}||_\infty + \gamma\beta(\tau\cdot C(Q_{t+1},\epsilon) + \epsilon)
\end{aligned} \tag{23}$$

where the last inequality follows from Lemma A.2 and $C(Q_{t+1},\epsilon)$ is given by

$$C(Q_{t+1},\epsilon) := \max_{s,z,a}(\int_{\mathcal{Z}} 1dz - 1 - \log\int_{\mathcal{C}(F_{s,z,a},\epsilon)} 1dz) \tag{24}$$

where $F_{s,z,a}(z') := \mathbb{E}_{s'\sim P_{s,z,a},a'\sim\pi(\cdot|s',z')}[Q_{t+1}(s',z',a')]$

Therefore

$$||Q_t - Q^*_{\mathrm{rob}}||_\infty \leq \gamma^t||Q_0 - Q^*_{\mathrm{rob}}||_\infty + \beta\sum_{k=1}^t\gamma^{t-k+1}(\tau\cdot C(Q_k,\epsilon) + \epsilon) \tag{25}$$

□

A.2. Why not merging states and contexts

We present a new theorem here to show that, under certain condition, our framework achieves tighter worst-case bound compared with the robust MDP merging context into state. Consider the uncertainty set which merges context into state and thus also takes the deviations of states transition into account:

$$\widetilde{\mathcal{U}}_{s,z,a} = \{(1-\beta)M_{s,z,a} + \beta\tilde{q}(s',z') \mid \tilde{q} \in \Delta(\mathcal{S} \times \mathcal{Z})\}, \tag{26}$$

where $M(s',z'|s,z,a) = \bar{P}(z'|z)P(s'|s,z,a)$ as in Eq. 1.

Let π_r^* be the optimal robust policy under uncertainty set $\widetilde{\mathcal{U}}$ and π_{rs}^* be the optimal robust policy under our uncertainty set $\mathcal{U}$ (Equation 2). Clearly, given β, we have $\mathcal{U}_{s,z,a} \subset \widetilde{\mathcal{U}}_{s,z,a}$ for all s, z, a.

For any policy π, let V_r^π be the robust value function of π under uncertainty set $\widetilde{\mathcal{U}}$ and V_{rs}^π be the robust value function of π under our uncertainty set $\mathcal{U}$.

Theorem A.3. *Assume the reward is normalized to the range $[-1,1]$. For any transition kernel $\hat{M} \in \mathcal{U}$, let $\hat{\pi}$ be the corresponding optimal policy in $\hat{M}$. Then we have*

(a)

$$V_{\hat{M}}^{\hat{\pi}}(\rho) - \frac{\beta}{1-\gamma} \le V_{\hat{M}}^{\pi_r^*}(\rho) \le V_{\hat{M}}^{\hat{\pi}}(\rho) \quad \text{and} \quad V_{\hat{M}}^{\hat{\pi}}(\rho) - \frac{\beta}{1-\gamma} \le V_{\hat{M}}^{\pi_{rs}^*}(\rho) \le V_{\hat{M}}^{\hat{\pi}}(\rho) \tag{27}$$

where $V_{\hat{M}}^{\pi}(\rho)$ is the expected discounted return under the transition kernel $\hat{M}$ and the initial distribution ρ, for policy π.

(b) *In general, $V_{rs}^{\pi_{rs}^*}(\rho) \ge V_r^{\pi_r^*}(\rho)$. Suppose $V_{rs}^{\pi_{rs}^*}(\rho) \ge V_r^{\pi_r^*}(\rho) + C$ for some constant $C \ge 0$, then*

$$V_{\hat{M}}^{\pi_{rs}^*}(\rho) \ge V_{\hat{M}}^{\pi_r^*}(\rho) + C - \frac{\beta}{1-\gamma} \tag{28}$$

Proof. We omit ρ here for convenience. (a) can be easily obtained from the usual robust MDP formulation. For (b),

$$V_{\hat{M}}^{\pi_{rs}^*} - V_{\hat{M}}^{\pi_r^*} = V_{\hat{M}}^{\pi_{rs}^*} - V_{rs}^{\pi_{rs}^*} + V_{rs}^{\pi_{rs}^*} - V_{\hat{M}}^{\pi_r^*} \ge V_{rs}^{\pi_{rs}^*} - V_{\hat{M}}^{\pi_r^*} = V_{rs}^{\pi_{rs}^*} - V_r^{\pi_r^*} + V_r^{\pi_r^*} - V_{\hat{M}}^{\pi_r^*} \ge C - \frac{\beta}{1-\gamma}$$

where the first inequality follows from the fact that $V_{\hat{M}}^{\pi_{rs}^*} \ge V_{rs}^{\pi_{rs}^*}$ by definition and the second inequality is due to $V_r^{\pi_r^*} - V_{\hat{M}}^{\pi_r^*} \ge -\frac{\beta}{1-\gamma}$ from the worst-case bound of π_r^* (since we also have $\hat{M} \in \widetilde{\mathcal{U}}$) and the assumption $V_{rs}^{\pi_{rs}^*}(\rho) \ge V_r^{\pi_r^*}(\rho) + C$.

□

The constant C represents the value estimation gap when one additionally considers the deviation of state transition. To achieve strict inequality $V_{\hat{M}}^{\pi_{rs}^*}(\rho) > V_{\hat{M}}^{\pi_r^*}(\rho)$, the theorem indicates that it is sufficient to have $C - \frac{\beta}{1-\gamma} > 0$. As an example the toy ACC (where $\beta = 0.2, \gamma = 0.99$), after normalizing the reward ($r = 0.1 \cdot (10-d)$ and otherwise -1 when the episode is early terminated), we can directly calculate that $V_{rs}^{\pi_{rs}^*} \approx 7.72$, $V_r^{\pi_r^*} \approx -16.86$. Then we can take $C = 24.58$ and thus $C - \frac{\beta}{1-\gamma} \approx 4.58 > 0$. This implies that our formulation achieves better performance compared with the robust MDP merging context into state, and thus the worst-case bound is tighter in this case.

B. Details of MuJoCo Experiments

B.1. Environment Details

HalfCheetah-acc and Hopper-acc. In these two environments, we augment the original MuJoCo state s_{MuJoCo} with two features, relative distance d and speed of lead car v, where the lead car is the reference frame to calculate d. The new state $s = (s_{\text{MuJoCo}}, d)$ and context $z = v$. At time step t, after doing an action, agent's velocity u_t changes to u_{t+1} by calling MuJoCo simulator and the transition of relative distance is

$$d_{t+1} = d_t + u_{t+1} \cdot \Delta t - v_t \cdot \Delta t. \tag{29}$$

Table 1. Specific hyperparameters for RS-SAC

Parameter	Value
β	0.3
τ	0.01
noise clip (c)	0.5
noise samples (K)	16

The context transition is $v_{t+1} = \bar{P}_{\Delta v}(v_t) = \max\{v_t + \Delta v, 0\}$, where $\Delta v \sim \mathcal{N}(\mu, \sigma)$ is the change of speed.

Let R be the original MuJoCo reward. Since the agent has to learn to walk first, we set the reward to be R when the agent is far from the lead car. Then we add a term related to the relative distance to encourage the agent to stay close to the lead car. Finally, when crashing ($d \geq 0$), reward becomes -10 to penalize the agent and the episode will be terminated. Specifically,

$$r(s,z,a) = \begin{cases} R(s_{\text{MuJoCo}}, a) & \text{if } d < -10 \\ R(s_{\text{MuJoCo}}, a) + 0.3 \cdot (10 + d) & \text{if } -10 \leq d < 0 \\ -10 & \text{if } 0 \leq d. \end{cases} \tag{30}$$

The maximum episode length is 500. The initial context is 0 and the initial relative distance is uniformly sampled from the interval $[-6, -5]$. The parameters μ and σ of training environment is set to be $\mu = 0$ and $\sigma = 0.2$.

Ant-cross. In this environment, the state is the same as the original MuJoCo state which includes the current position of the agent. The moving obstacle is of radius 0.5. Its x-position is fixed to be 2 and its y-position y_t is the context. The context transition here is i.i.d. $y_t \sim \mathcal{N}(\mu, \sigma)$ and we clip y_t to be the range $[-0.4, 0.4]$. The goal position is set to be located at $(4, 0)$.

Let d_o be the distance between the agent and the obstacle, and let d_g be the distance between the agent and the goal position. The reward function is defined as

$$r(s,z,a) = \begin{cases} (4 - d_g) + (0.2 \cdot v_x + 0.4 \cdot |v_y|) - 0.01 \cdot \text{control\_cost} & \text{if } 0 < d_o \\ -10 & \text{if } d_o \leq 0. \end{cases} \tag{31}$$

where v_x and v_y are the x and y velocity of the agent, respectively. The first term $4 - d_g$ is dominant, in order to let the agent stay close to the goal position. The second term $0.2 \cdot v_x + 0.4 \cdot |v_y|$ is to encourage the agent to go right and move up or down to avoid the moving obstacle. We balance the control cost by multiplying a small constant in the third term.

The episode will be terminated if $d_o \leq 0$. The maximum episode length is 500. The initial context is 0 and the initial state distribution is the same as that of MuJoCo. The parameters μ and σ of training environment is set to be $\mu = 0$ and $\sigma = 0.3$.

B.2. Implementation Details

Algorithm Implementations. SAC is implemented following the original SAC paper (Haarnoja et al., 2018). SC-SAC is implemented with state noise $\epsilon = 0.005$, same as that in original paper (Kuang et al., 2022). We implement PR-SAC following (Tessler et al., 2019) and (Kuang et al., 2022) where the adversary policy is TD3-type (Fujimoto et al., 2018) and we set the probability of the adversary to be 0.1 and the training frequency of the adversary to be $10 : 1$. The hyperparameters for the softmin operator in RS-SAC described in Section 3.2 are listed in Table 1.

For HalfCheetah-acc and Hopper-acc, DR-SAC is trained on environments with parameters sampled uniformly from $\mu \in [-0.2, 0]$, $\sigma \in [0.2, 0.6]$. For Ant-cross, DR-SAC is trained on environments with parameters sampled uniformly from $\mu \in [-0.2, 0.2]$, $\sigma \in [0.3, 0.6]$.

Shared Hyperparameters. Since all algorithms are SAC-type, they share the same hyperparameters as those in original SAC (Haarnoja et al., 2018), shown in Table 2.

Evaluation Settings. For HalfCheetah-acc and Hopper-acc, we change environment parameters μ and σ to be in the ranges $\mu \in [-0.4, 0.1]$ and $\sigma \in [0.1, 2.0]$. Note that when μ is greater than 0, the speed of lead car can easily go beyond the maximum velocity of the agent which is limited by the physics simulation of MuJoCo. Thus when $\mu > 0$, all algorithms cannot achieve high returns as shown in Figure 3. For Ant-cross, environment parameters μ and σ are changed to be in the ranges $\mu \in [-0.3, 0.3]$ and $\sigma \in [0.2, 0.7]$.

Table 2. Shared hyperparameters for all algorithms

Parameter	Value
number of hidden layers	2
number of units per layer	256
activation	RELU
optimizer	Adam
discount factor	0.99
learning rate	$3 \cdot 10^{-4}$
replay buffer size	10^6
batch size	256
target entropy	$-\dim(\mathcal{A})$
soft update coefficient	$5 \cdot 10^{-3}$
soft update interval	1
update every	1

B.3. Training Curves in MuJoCo Tasks

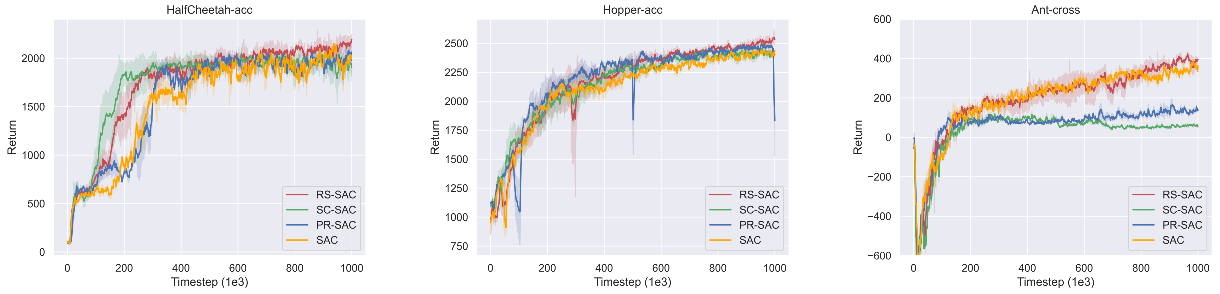


Figure 7. Training curves in MuJoCo tasks with dynamic contexts

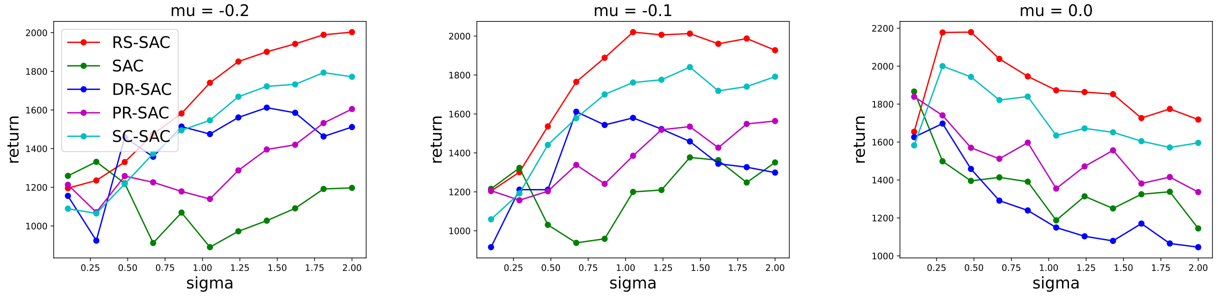
Figure 7 shows that all algorithms achieve similar training performances in HalfCheetah-acc and Hopper-acc, while in Ant-cross, PR-SAC and SC-SAC even lead to performance degradation. This is mainly due to the difficulty of Ant-cross, where the agent obtains reward signal about the obstacle only when hitting to it and so this signal is rather sparse. Thus optimizing over worst-case disturbances in state-context space, like SC-SAC, or action space, like PR-SAC, might be very hard. We do not plot the training curve of DR-SAC since it is trained on multiple environments with different parameters while others use only one nominal training environment.

B.4. More Details about Figure 3

Here we fix several values for μ and plot the average return curve for each algorithm as a function of σ in our MuJoCo tasks for a clearer visualization. For HalfCheetah-acc and Hopper-acc, we choose $\mu = -0.2, -0.1, 0$, see Figure 8 and 9. For Ant-cross, we choose $\mu = -0.12, 0, 0.12$, see Figure 10.

B.5. Ablation Study

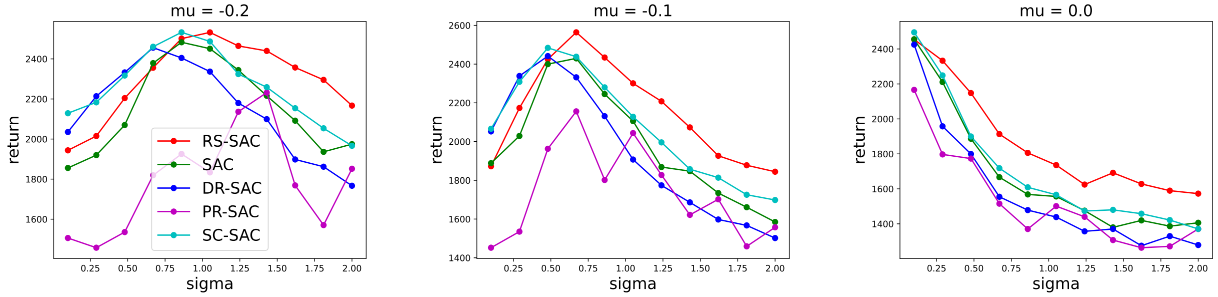
We conduct ablation experiments on HalfCheetah-acc to study the effects given by different noise clip parameters $c = 0.01, 0.05, 0.1, 0.5$ (original), $1.0, 1.2$, see Figure 11. Notice that when $c > 1$, the performance degenerates quickly, since searching worst-case contexts over large ranges leads to over-pessimism in value estimation. While when c is small ($c = 0.01$ or 0.05), the performance reduces to be non-robust as the robust critic update tends to the non-robust one due to the small range.


mu = -0.2

mu = -0.1

mu = 0.0

Figure 8. Performances in several target context transitions (μ, σ) for HalfCheetah-acc.


mu = -0.2

mu = -0.1

mu = 0.0

Figure 9. Performances in several target context transitions (μ, σ) for Hopper-acc.

C. Details of Inventory Control Task

Environment Details. The state-context space consists of the incoming orders of dimension equal to lead time, the in-stock levels and demands. The action space is of one dimension, deciding how many order to place which is normalized to the range $[0, 4]$. Our goal is to maximize the profit and minimize the cost originated from restocking, backlogs, etc. Specifically, profit is equal to the fixed price times number of item sold, denoted by n_sold. The overall inventory cost, denoted by inventory_cost, is the product of current in-stock level and inventory cost per item per day. The restocking cost, denoted by restock_cost, is the cost for restock one item times the number of orders, with additional fixed cost for restocking added. The backlog cost, denoted by backlog_cost, is proportional to the number of demands that are not met by the inventory since then customers tend to go to competitors. We list the related constants to the Table 3. Finally, we take reward r to be a linear combination of these terms, which is defined as $r = \text{income} - \text{inventory\_cost} - \text{restock\_cost} - \text{backlog\_cost}$.

The initial in-stock level is set to be 20 and initial demand is uniformly sampled from the dataset. Each episode is of length 100.

Implementation Details. We keep hyperparameters of algorithms in inventory control same as those in MuJoCo tasks with dynamic contexts, except that for RS-SAC, where the critical parameter β is set to be 0.01. The reason is that the context transition in inventory control is highly stochastic, unlike the hand-crafted one in MuJoCo tasks, and so RS-SAC are hard to optimize with large β and sensitive to β. We find that $\beta = 0.01$ achieves good performance for RS-SAC.

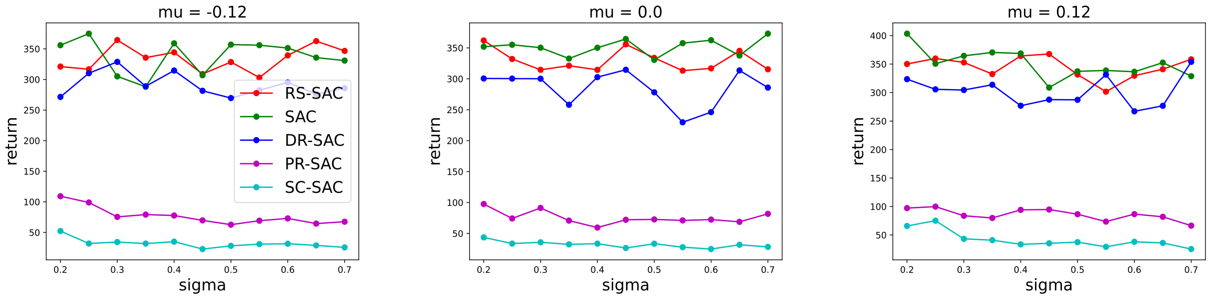


Figure 10. Performances in several target context transitions (μ, σ) for Ant-cross.

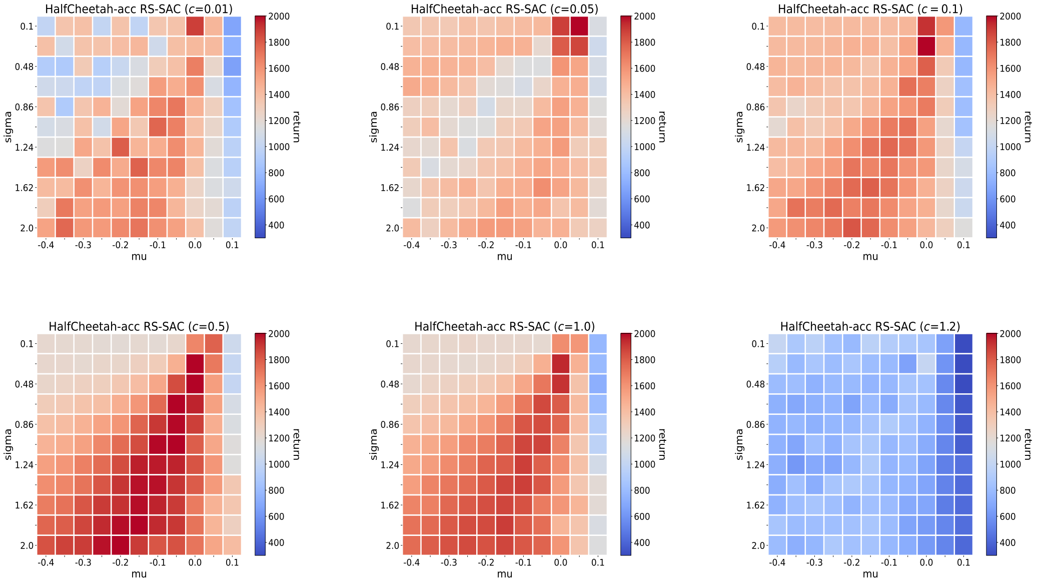


Figure 11. Ablation study in HalfCheetah-acc.

Table 3. Environment parameters for inventory control

Parameter	Value
cost for restocking	60
cost for restock one item	1
inventory cost per item per day	0.1
price per item	3
backlog cost per item	1
lead time	7