ACTOR-CRITIC ALGORITHM FOR DYNAMIC EXPECTILE AND CVAR

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimizing dynamic risk with stochastic policies is challenging in both policy updates and value learning. The former typically requires transition perturbation, while the latter may rely on model-based approaches. To address these challenges, we propose a surrogate policy gradient without transition perturbation under softmax policy parameterization. We further develop model-free value learning methods for dynamic expectile and conditional value-at-risk (CVaR) by leveraging elicitability. Finally, inspired by Expected SARSA and Expected Policy Gradient, the first model-free off-policy actor-critic algorithm with stochastic policies is constructed for dynamic expectile and CVaR. Empirical results in domains with verifiable risk-averse behavior show that our algorithm can learn risk-averse policy and consistently outperforms other existing methods.

1 INTRODUCTION

The demand for risk control in sequential decision-making has inspired risk-averse reinforcement learning (RL). Generally, there exist two main types of risk-sensitive objectives in the literature. One defines risk over the total return variable and is referred to as static risk. The other defines the risk recursively along the trajectory and is referred to as dynamic risk. While static risk has advantages in interpretability, optimizing static risk can be difficult as its optimal policy is generally history-dependent (Benyamine et al., 2026). In contrast, dynamic risk models the behavior of a decision maker who accounts for risk at each step. Its recursive structure also leads to a Bellman-type equation, which implies the optimal policy is Markovian (Ruszczyński, 2010).

Although the Bellman equation for dynamic risk is analogous to the risk-neutral case, developing a practical actor-critic algorithm raises challenges for two reasons. 1) For coherent risk measures (Artzner et al., 1999), computing policy gradients with stochastic policy requires perturbing the transition probabilities according to its risk envelope (Tamar et al., 2015; Zhang et al., 2024), which significantly increases complexity. 2) The value function is itself a risk measure over the randomness induced by the transition dynamics, which is usually hard to obtain. Extracting the risk value directly from the value distribution learned by distributional RL (Dabney et al., 2018) does not guarantee correctness (Lim & Malik, 2022). As a result, most existing works may either adopt model-based approaches for value learning (e.g.,Tamar et al. (2015); Rigter et al. (2023)) or resort to deterministic policy gradient to simplify policy updates (e.g.,Marzban et al. (2023)).

We focus on two widely used coherent risk measures for dynamic risk, namely expectile and conditional value-at-risk (CVaR), and aim to develop model-free actor-critic algorithms with stochastic policies for them. First, from a policy iteration perspective, we show for the first time that a stochastic policy updated by a surrogate policy gradient, without perturbing transition probabilities, converges to the optimal policy with proper learning rate, when using the popular softmax parameterization. Next, leveraging the elicitability of expectile (Marzban et al., 2023) and motivated by the joint elicitability of CVaR (Coache et al., 2023), we derive value learning rules based on stochastic optimization, analogous to Q-learning in the risk-neutral setting. Finally, inspired by Expected SARSA (Sutton et al., 1998) and Expected Policy Gradient (Ciosek & Whiteson, 2020), an off-policy actor-critic algorithm is formalized. To the best of our knowledge, this is the first model-free off-policy actor-critic framework with stochastic policies for dynamic expectile and CVaR, and the idea of policy improvement also applies to other dynamic risk measures. Empirical results in do-

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

mains with verifiable risk-averse behavior demonstrate that our methods can learn risk-averse policy when others may fail.

2 PRELIMINARY

2.1 RISK MEASURES, EXPECTILE, AND CVAR

For a real valued random variable $\tilde{x}$ (we use tilde to mark a random variable, e.g., $\tilde{x}$, in this paper), a *risk measure* is a function $\rho : \tilde{x} \to \mathbb{R} \cup \{-\infty, +\infty\}$ that maps an uncertain return to a quantity that captures the risk adjusted value of $\tilde{x}$. An important class is *coherent* risk measure, that possesses a set of properties that are consistent with rational risk assessments (Artzner et al., 1999), i.e.,

A1 Concavity: $\forall \lambda \in [0, 1], \rho(\lambda\tilde{x} + (1-\lambda)\tilde{y}) \geq \lambda\rho(\tilde{x}) + (1-\lambda)\rho(\tilde{y})$;

A2 Monotonicity: if $\tilde{x} \leq \tilde{y}$, then $\rho(\tilde{x}) \leq \rho(\tilde{y})$;

A3 Translation invariance: $\forall c \in \mathbb{R}$, $\rho(\tilde{x} + c) = \rho(\tilde{x}) + c$;

A4 Positive homogeneity: if $\lambda \geq 0$, then $\rho(\lambda\tilde{x}) = \lambda\rho(\tilde{x})$.

Superadditivity (i.e., $\rho(\tilde{x}+\tilde{y}) \geq \rho(\tilde{x})+\rho(\tilde{y})$) can be inferred from (A1) and (A4), which may appear in the definition of coherence in other papers. Note that since we work on return instead of cost, some properties, i.e. Concavity and Superadditivity, are adjusted accordingly. Here, we consider two common coherent risk measures, i.e., expectile and CVaR.

For a risk level $\alpha \in (0, 1)$, the α-expectile is defined as (Bellini & Di Bernardino, 2017)

$$\text{Expectile}_\alpha[\tilde{x}] := \arg\min_{y \in \mathbb{R}} (1-\alpha)\mathbb{E}\big[(\tilde{x} - y)_-^2\big] + \alpha\mathbb{E}\big[(\tilde{x} - y)_+^2\big], \tag{1}$$

where $(x)_- := \min(x, 0)$ and $(x)_+ := \max(x, 0)$. Expectile covers the case of risk-neutrality (i.e., expectation) when $\alpha = 1/2$. As $\alpha \to 0$ or $\alpha \to 1$, it recovers $\operatorname{ess\,inf}(\tilde{x})$ or $\operatorname{ess\,sup}(\tilde{x})$, respectively.

The left-tail CVaR is defined as (Rockafellar et al., 2000)

$$\text{CVaR}_\alpha[\tilde{x}] := \max_{y \in \mathbb{R}} y - \frac{1}{\alpha}\mathbb{E}[(y - \tilde{x})_+], \tag{2}$$

where the maximum is always attained at y equals the α-value-at-risk (VaR) of $\tilde{x}$ as a by product. The α-VaR is defined as the largest $1-\alpha$ confidence lower bound on the value of $\tilde{x}$, i.e., $\text{VaR}_\alpha[\tilde{x}] := \max\{x \in \mathbb{R} | P[\tilde{x} < x] \leq \alpha\}$. For ease of presentation, we assume the random variable is continuous and its quantile is unique in this paper. In this case, VaR coincides with quantile, and CVaR can be alternatively expressed as $\text{CVaR}_\alpha[\tilde{x}] := \frac{1}{\alpha}\int_0^\alpha \text{VaR}_\beta[\tilde{x}]d\beta$. Thus, $\text{CVaR}_\alpha[\tilde{x}]$ can be interpreted as the expected value of the bottom α portion of tail quantiles, i.e., $\text{CVaR}_\alpha[\tilde{x}] = \mathbb{E}[\tilde{x} | \tilde{x} \leq \text{VaR}_\alpha[\tilde{x}]]$. As $\alpha \to 0$ or $\alpha \to 1$, it recovers $\operatorname{ess\,inf}(\tilde{x})$ or $\mathbb{E}[\tilde{x}]$, respectively.

2.2 (COHERENT) DYNAMIC RISK IN RL

In standard RL settings, the interaction between the agent and the environment is modeled as a MDP, represented by a tuple $\langle \mathcal{S}, \mathcal{A}, R, P, \mu_0, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote state and action spaces. $R(s, a, \tilde{\omega})$ is the state and action dependent reward function with randomness $\tilde{\omega}$. $P(\cdot|s, a)$ is the transition function. μ_0 is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. For notational simplicity, we use $\tilde{r}(s, a)$ to denote the random reward variable associated with the pair (s, a) and $r(s, a)$ to denote a realization. When the context is clear, we may drop the (s, a) argument and simply use $\tilde{r}$ or r. The agent collects a sequence of state $\tilde{s}_t$, action $\tilde{a}_t$, and reward $\tilde{r}_t$ by executing its (Markovian) policy π in the MDP. Without loss of generality, we consider s_0 is fixed when defining the objective. For simplicity, we begin from the case where π is deterministic to introduce policy learning of dynamic risk, followed by the discussion for the case where π is stochastic. We overload the notation for π, so that $\pi(s)$ refers to a deterministic policy and $\pi(\cdot|s)$ refers to a stochastic one.

2.2.1 DETERMINISTIC POLICY CASE

For a policy π, MDP M, coherent risk measure ρ (e.g.,α-expectile or α-CVaR), the dynamic risk objective $\rho_\infty^\pi(M)$ is defined as (Rigter et al., 2023)

$$\rho_\infty^\pi(M) = \rho\Big(\tilde{r}(s_0, \pi(s_0)) + \gamma\rho\big(\tilde{r}(\tilde{s}_1, \pi(\tilde{s}_1)) + \gamma\rho(\tilde{r}(\tilde{s}_2, \pi(\tilde{s}_2)) + ...)\big)\Big). \tag{3}$$

Define the action-value function under policy π as $Q^\pi(s,a) = \rho^\pi_\infty(M|s_0 = s, a_0 = a)$, and the state-value function as $V^\pi(s) = \rho^\pi_\infty(M|s_0 = s)$ (it holds that $V^\pi(s) = Q^\pi(s, \pi(s))$). The Bellman equation is given by (Ruszczyński, 2010)

$$Q^\pi(s,a) = \rho\big(\tilde{r}(s,a) + \gamma Q^\pi(\tilde{s}', \pi(\tilde{s}'))\big). \tag{4}$$

Solving for $\max_\pi \rho^\pi_\infty(M)$ admits an optimal value function Q^*, where $Q^*(s,a) = \max_\pi Q^\pi(s,a)$, $V^*(s) = \max_a Q^*(s,a)$, and the corresponding optimal policy is $\pi^*(s) = \arg\max_a Q^*(s,a)$. The Bellman optimality equation is (Ruszczyński, 2010)

$$Q^*(s,a) = \rho\big(\tilde{r}(s,a) + \gamma \cdot \max_{a'} Q^*(\tilde{s}', a')\big) \tag{5}$$

When ρ is $\mathbb{E}$, these two equations reduce to the standard Bellman equations in risk-neutral setting. For general (non-linear) risk measures, computing ρ in a closed is often difficult. A straightforward approach is to learn environment dynamics (i.e., model-based) to approximate the distribution of $\tilde{r}$ and $\tilde{s}'$, so as to assist computing ρ (e.g., Rigter et al. (2023) for dynamic CVaR). For clarity of presentation, we defer the discussion of model-free value learning for dynamic expectile and CVaR to Sec. 3.2. For now, we assume $Q^\pi(s,a)$ can be learned via Eq. 4 for any π.

Though Eq. 5 suggests a value-based approach similar to Q-learning (Sutton et al., 1998), actor-critic methods are often preferred for better stability and generalization. Let π be parameterized by θ. When π is deterministic, the construction of an actor-critic is similar to deterministic policy gradient (DPG) (Silver et al., 2014) in risk-neutral case (e.g.,Marzban et al. (2023) for dynamic expectile). The policy update takes the form $\int_S d(s)\nabla_\theta \pi_\theta(s) \nabla_a Q^\pi(s,a)|_{a=\pi_\theta(s)} ds$, where $d(s)$ is a state distribution. However, DPG is mainly designed for continuous action spaces, which makes it inapplicable when actions are discrete.

2.2.2 Stochastic Policy Case

For stochastic policies, there exist two ways to define the risk objective as in Eq. 3, depending on how the randomness induced by the policy is handled. One incorporates the randomness of $\pi(\cdot|s)$ directly into ρ, so that the risk is taken over the joint distribution of $\tilde{s}$, $\tilde{a}$, and $\tilde{r}$ (Tamar et al., 2015; 2016; Huang et al., 2021), i.e., $\rho^\pi_\infty(M) = \rho\big(\tilde{r}(s_0,\tilde{a}_0) + \gamma\rho(\tilde{r}(\tilde{s}_1,\tilde{a}_1) + ...)\big)$. The other marginalizes the policy randomness by taking expectation (Shen et al., 2013; Zhang et al., 2024), i.e.,

$$\rho^\pi_\infty(M) = \mathbb{E}_{a_0\sim\pi(\cdot|s_0)}\big[\rho\big(\tilde{r}(s_0,a_0) + \gamma \mathbb{E}_{a_1\sim\pi(\cdot|\tilde{s}_1)}[\rho(\tilde{r}(\tilde{s}_1,a_1) + ...)]\big)\big], \tag{6}$$

with unique fixed point Q^π and with $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)]$. In addition, Zhang et al. (2024) equivalently reformulates the dynamic risk defined in Eq. 6 as a soft-robust MDP model and highlights the challenges in deriving proper model-free stochastic policy gradient estimator (see Remark 4 therein). In this work, we adopt the latter definition for two reasons. 1) Including policy stochasticity in the risk measure treats exploration as a source of risk, which may hinder exploration in risk-averse setting. 2) Since there exists optimal deterministic policy (Eq. 5), a stochastic policy is expected to converge toward a deterministic one. Accordingly, we consider the following Bellman operator for stochastic policies, similar to Zhang et al. (2024)

$$\mathcal{T}^\pi Q(s,a) = \rho\big(\tilde{r}(s,a) + \gamma \sum_{a'} \pi(a'|\tilde{s}')Q(\tilde{s}', a')\big) \tag{7}$$

Next, consider the policy update. By the dual representation of coherent risk measures, $\rho(\cdot)$ can be expressed as an expectation under a perturbed distribution. This allows one to extend stochastic policy gradient (SPG) methods (Sutton et al., 1998) from the risk-neutral setting to dynamic risk. Specifically, the dual form is $\rho(\tilde{x}) = \inf_\xi \mathbb{E}_\xi[\tilde{x}]$, where ξ is the risk envelope satisfying $\int_x \xi(x) f_{\tilde{x}}(x) = 1$ and $\xi(x) \geq 0$. The additional constraints on ξ depend on the specific choice of ρ. In Eq. 7, the randomness comes jointly from $\tilde{r}$ and $\tilde{s}'$. As a result, ξ perturbs the transition distribution $P(r, s'|s,a)$ to $\xi(r, s'|s,a)P(r, s'|s,a)$. Let ξ^* denote the optimal risk envelope and P^{ξ^*} the corresponding perturbed transition. The policy gradient is then given by (Zhang et al., 2024)

$$\nabla_\theta V^{\pi_\theta}(s_0) = \mathbb{E}_{a_0\sim\pi_\theta(\cdot|s_0)}\mathbb{E}_{\xi^*}\Big[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_\theta}(s_t,a_t)\nabla_\theta \log \pi_\theta(a_t|s_t)\Big], \tag{8}$$

where $\mathbb{E}_{\xi^*}$ denotes that the trajectory $\{s_t, a_t, r_t\}$ follows the perturbed transition P^{ξ^*}. The derivation is provided in Appendix A.1. Note that when the risk measure also accounts for policy randomness, the corresponding policy gradient similarly involves transition perturbation (e.g., see Tamar et al. (2015)). However, computing ξ^* at each step is non-trivial, as it typically requires solving a constrained optimization problem, often via Lagrangian methods (Tamar et al., 2015).

3 METHOD

As discussed in Sec. 2.2, applying SPG to coherent dynamic risk generally requires solving for the optimal transition perturbation at each step, which may be difficult to compute and implement in practice. Although DPG avoids this issue, it is limited to continuous action spaces. Therefore, a model-free actor-critic algorithm with stochastic policies is desired. In this section, we show that this can be achieved by using a surrogate policy gradient that avoids transition perturbation.

3.1 A SURROGATE STOCHASTIC POLICY GRADIENT

We first claim the assumptions we made for tabular MDP.

Assumption 3.1. *1) The space $\mathcal{S} \times \mathcal{A}$ is finite and $d(s)$ is a normalized state distribution satisfying $0 < d(s) \leq 1$. 2) Reward $\tilde{r} \in [0, 1]$. 3) The reward has bounded densities $\forall(s, a)$.*

With assumtion 3.1, the surrogate gradient we consider for policy parameter θ is

$$\theta_{t+1} \leftarrow \theta_t + \eta \sum_s d(s) \sum_a \nabla_\theta \pi_{\theta_t}(a|s) Q^{\pi_t}(s, a), \tag{9}$$

where θ_t is the policy parameter at iteration t, and η is the learning rate. Here, $Q^{\pi_t} = Q^{\pi_{\theta_t}}$ is treated as fixed. This gradient formulation is analogous to SPG in risk-neutral setting (Sutton et al., 1998) but it uses risk-sensitive Q-function. Moreover, Eq. 9 does not require transition perturbation as it only requires the state distribution $d(s)$ to have full coverage. We assume that Q^{π_t} (Eq. 7) is available under π_{θ_t} when analyzing the behavior of policy update. This follows the standard policy iteration view, where the value function is fully evaluated before updating the policy.

The motivation of proposing Eq. 9 is based on the well known connection between DPG and SPG in risk-neutral setting, namely that DPG is the limit of SPG as policy variance goes to zero (Silver et al., 2014; Ciosek & Whiteson, 2020). Recall that DPG provides a model-free approach to learning dynamic risk (Sec. 2.2.1). This connection therefore motivates us to investigate the behavior of risk-neutral-style SPG (Eq. 9) in risk-sensitive setting. Here, we are interested in the popular softmax policy parameterization when action is discrete, i.e., $\pi_\theta(a|s) = \exp\big(\theta(s, a)\big) / \sum_b \exp\big(\theta(s, b)\big)$. In this case, the update rule for $\theta_t(s, a)$ can be simplified as $\theta_{t+1}(s, a) \leftarrow \theta_t(s, a) + \eta\, d(s) \pi_t(a|s) A^{\pi_t}(s, a)$, where $A^{\pi_t}(s, a) = Q^{\pi_t}(s, a) - V^{\pi_t}(s)$ (see Appendix A.2). With softmax parameterization, Eq. 9 (with proper learning rate) ensures the monotone increasing of value function and the convergence to the optimal value (therefore an optimal policy), as formally stated in Theorem 3.2 and Theorem 3.3.

Theorem 3.2. *With assumption 3.1, for softmax policy, the update rule in Eq. 9 with $\eta \leq (1 - \gamma)/5$ ensures that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $Q^{\pi_{t+1}}(s, a) \geq Q^{\pi_t}(s, a)$, $V^{\pi_{t+1}}(s) \geq V^{\pi_t}(s)$.*

Theorem 3.3. *With assumption 3.1 and the same conditions as in Theorem 3.2, the update rule in Eq. 9 ensures that for all s, $V^{\pi_t}(s) \to V^*(s)$ as $t \to \infty$.*

The proofs of Theorem 3.2 and Theorem 3.3, which are presented in Appendix B.1 and B.2, carefully adapt the arguments of Agarwal et al. (2021) to the risk-averse setting. The key challenge in our proof resides in circumventing the fact that the gradient in Eq. 9 can no longer be interpreted as the gradient of the value function in a risk-averse setting (see detailed discussion of differences of arguments in Remark B.9). The assumption that $\tilde{r} \in [0, 1]$ is for ease of presentation without loss of generality. For proving Theorem 3.2, the key idea is that with proper learning rate η, $\mathbb{E}_{a \sim \pi_{t+1}(\cdot|s)}[Q^{\pi_t}(s, a)] \geq \mathbb{E}_{a \sim \pi_t(\cdot|s)}[Q^{\pi_t}(s, a)]$. By the monotonicity (A2) of ρ in Eq. 7, this implies $\mathcal{T}^{\pi_{t+1}} Q^{\pi_t} \geq \mathcal{T}^{\pi_t} Q^{\pi_t} = Q^{\pi_t}$. Applying monotonicity again yields $Q^{\pi_{t+1}} = \lim_{k \to \infty} (\mathcal{T}^{\pi_{t+1}})^k Q^{\pi_t} \geq \mathcal{T}^{\pi_{t+1}} Q^{\pi_t} \geq Q^{\pi_t}$. Since the value function is pointwise increasing and bounded, the monotone convergence theorem ensures the existence of some Q^∞ and V^∞ such that $Q^{\pi_t}(s, a) \to Q^\infty(s, a)$ and $V^{\pi_t}(s) \to V^\infty(s)$. For Theorem 3.3, define the action sets $I_s^0 := \{a | Q^\infty(s, a) = V^\infty(s)\}, I_s^+ := \{a | Q^\infty(s, a) > V^\infty(s)\}, I_s^- := \{a | Q^\infty(s, a) < V^\infty(s)\}$. The proof steps are to show $\pi_t(a|s) \to 0$ for actions $a \in I_s^+ \cup I_s^-$ as $t \to \infty$, and I_s^+ is empty. The first part is based on the fact that the gradient tends to zero in the limit while $|A^{\pi_t}(s, a)|$ for $a \in I_s^+ \cup I_s^-$ is non-zero. $I_s^+ = \emptyset$ is obtained by contradiction. If I_s^+ is non-empty, then one can show that there is a subset set of actions in I_s^0 for which the summation of $\theta(s, a)$ goes to infinity while the summation of their updates is negative, thus leading to a contradiction.

For continuous action, using common Gaussian policy is a reasonable heuristic given the connection between SPG and DPG (Silver et al., 2014). As a result, Eq. 9 suggests a model-free approach to

policy updates for dynamic risk using stochastic policies, which avoids transition perturbation and bypasses the learning of optimal risk envelope, as required in many existing methods (Tamar et al., 2015; 2016; Huang et al., 2021; Zhang et al., 2024). This surrogate gradient is central to the actor-critic algorithm proposed in this paper.

3.2 Learning Value Function

So far, we have assumed that Q^π is available. In practice, learning Q^π from Eq. 7 is challenging when ρ is non-linear. Based on the idea of elicitability of expectile, and motivated by the joint elicitability of CVaR, we show how to learn value function from transition samples for them.

Dynamic Expectile. A risk measure ρ is elicitable if there exists a measurable function L (called a loss function) such that $\rho(\tilde{x}) = \arg\min_y \mathbb{E}[L(\tilde{x}, y)]$ (Bellini & Bignozzi, 2015). Expectile is the only coherent risk measure that is elicitable, with its loss function given in Eq. 1. Therefore, when ρ is α-expectile, Eq. 7 can be equivalently expressed as

$$\mathcal{T}^\pi Q(s,a) = \arg\min_y (1-\alpha)\mathbb{E}\big[(\tilde{r}+\gamma V(\tilde{s}')-y)_-^2\big] + \alpha\mathbb{E}\big[(\tilde{r}+V(\tilde{s}')-y)_+^2\big],$$

where $V(s') = \mathbb{E}_{a'\sim\pi(\cdot|s')}[Q(s',a')]$. This leads to a gradient-based Q-learning–style update, where Q is updated along the negative gradient of the loss. Given a transition (s,a,r,s') and learning rate ζ, the update is

$$\begin{aligned} Q_{t+1}(s,a) \leftarrow & Q_t(s,a) - \zeta_t(s,a)\cdot\partial_y\big[(1-\alpha)(r+\gamma V_t(s')-y)_-^2 + \alpha(r+\gamma V_t(s')-y)_+^2\big]\big|_{y=Q_t(s,a)} \\ = & Q_t(s,a) + 2\zeta_t(s,a)\cdot\big[(1-\alpha)(r+\gamma V_t(s')-Q_t(s,a))_- + \alpha(r+\gamma V_t(s')-Q_t(s,a))_+\big]. \end{aligned} \tag{10}$$

This model-free approach for learning value function under dynamic expectile was initially proposed by Marzban et al. (2023) yet used within a DPG algorithm without any convergence analysis.

Dynamic CVaR. CVaR is not elicitable, making it more challenging than expectile. However, it is jointly elicitable with VaR, i.e., $\big(\mathrm{VaR}_\alpha(\tilde{x}), \mathrm{CVaR}_\alpha(\tilde{x})\big) = \arg\min_{y_1,y_2}\mathbb{E}[L(\tilde{x},y_1,y_2)]$. L is not unique here, with a general form given in Fissler & Ziegel (2016). Let $q(s,a)$ denote the α-VaR action-value function. Based on joint elicitability, Coache et al. (2023) proposed to learn both $q(s,a)$ and $Q(s,a)$ simultaneously by minimizing $\mathbb{E}[L(\tilde{r}+\gamma V(\tilde{s}'), q(s,a), Q(s,a)]$. It is a challenging task due to the non-convexity of L and the fact that the partial derivatives of L involve compound terms that may hinder stability (see discussion in Appendix A.3). To address this, we instead propose a two-time-scale learning scheme. Namely, while VaR is elicitable via quantile regression, CVaR can be expressed as an expectation given VaR (Eq. 2), which leads to

$$\begin{aligned} q(s,a) &= \arg\min_{y_1}\mathbb{E}[l_\alpha(\tilde{r}+\gamma V(\tilde{s}')-y_1)],\ l_\alpha(x-y) := (\alpha - \mathbb{I}\{x<y\})(x-y) \\ Q(s,a) &= \arg\min_{y_2}\frac{1}{2}\big(y_2 - q(s,a) + \frac{1}{\alpha}\mathbb{E}[(q(s,a)-\tilde{r}-\gamma V(\tilde{s}'))_+]\big)^2, \end{aligned}$$

where $V(s') = \mathbb{E}_{a'\sim\pi(\cdot|s')}[Q(s',a')]$. Similar to Eq. 10, this formulation leads to stochastic gradient-based Q-learning-style updates for both q and Q. Given a transition (s,a,r,s') and learning rates ζ^q and ζ^Q (with $\zeta^q \gg \zeta^Q$, thus Q is quasi-static w.r.t. q), the updates are

$$\begin{aligned} q_{t+1}(s,a) &\leftarrow q_t(s,a) - \zeta_t^q(s,a)\cdot(\mathbb{I}\{r+\gamma V_t(s') < q_t(s,a)\} - \alpha) \\ Q_{t+1}(s,a) &\leftarrow Q_t(s,a) - \zeta_t^Q(s,a)\cdot(Q_t(s,a) - q_t(s,a) + \frac{1}{\alpha}(q_t(s,a)-r-\gamma V_t(s'))_+) \end{aligned} \tag{11}$$

The following propositions show that the value learning rules above enjoy convergence guarantees that are comparable to those in standard Q-learning. Proposition 3.4 follows from Shen et al. (2013), exploiting the Robbins-Monro conditions as in risk-neutral Q-learning (Bertsekas & Tsitsiklis, 1996), whereas Proposition 3.5 is new and employs the two-time scale framework in Borkar (2023), and the proof is to show the update satisfies the conditions of the framework.

Proposition 3.4. *With assumption 3.1, for Eq. 10, if the nonnegative learning rates $\zeta_t(s,a)$ satisfy $\sum_{t=0}^\infty \zeta_t(s,a) = \infty$ and $\sum_{t=0}^\infty \zeta_t^2(s,a) < \infty$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$, then $Q_t(s,a)$ converges to $Q^\pi(s,a)$ associated with dynamic expectile for all (s,a) almost surely.*

Proposition 3.5. *With assumption 3.1 and assuming that the sequences of q_t and Q_t are bounded almost surely, for Eq. 11, if the nonnegative learning rates $\zeta_t^q(s,a), \zeta_t^Q(s,a)$ satisfy $\sum_{t=0}^\infty \zeta_t^q(s,a) = \sum_{t=0}^\infty \zeta_t^Q(s,a) = \infty$, $\sum_{t=0}^\infty[\zeta_t^q(s,a)]^2, \sum_{t=0}^\infty[\zeta_t^Q(s,a)]^2 < \infty$, and $\lim_{t\to\infty}\frac{\zeta_t^Q(s,a)}{\zeta_t^q(s,a)} = 0$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$, then $Q_t(s,a)$ converges to $Q^\pi(s,a)$ associated with dynamic CVaR for all (s,a) almost surely.*

3.3 Actor-Critic Algorithm

It is now convenient to construct actor-critic algorithm with stochastic policies for dynamic expectile and CVaR based on Sec. 3.1 and Sec. 3.2. For policy updates, a common way to estimate the gradient in Eq. 9 is via Monte Carlo sampling, i.e., trajectories $\{s_t, a_t, r_t\}$ are generated by executing $\pi(\cdot|s)$ in the environment, and the collected samples are subsequently used to update both policy and value function. These trajectories are discarded after each policy update (if no importance sampling is applied). This is called on-policy policy gradient (PG) as in e.g., REINFORCE with baseline (Sutton et al., 1998) and PPO (Schulman et al., 2017). In essence, on-policy PG estimates the expectation over actions in Eq. 9 (since $\nabla_\theta \sum_a \pi_\theta(a|s) Q^\pi(s,a) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[Q^\pi(s,a)\nabla_\theta \log \pi_\theta(a|s)]$) by sampling a single action from $\pi(\cdot|s)$. In contrast, Expected Policy Gradient (EPG) (Ciosek & Whiteson, 2020) exploits the structure of discrete action spaces to compute $\nabla_\theta \sum_a \pi_\theta(a|s) Q^\pi(s,a)$ exactly by enumerating all $a \in \mathcal{A}$ at a state s. By explicitly integrating over the entire action distribution, the resulting gradient no longer depends on the particular action executed in the environment. As a result, EPG enables policy updates using states drawn from arbitrary distributions, such as those in an experience replay buffer (i.e., off-policy), effectively decoupling policy learning from the behavior distribution. For value function learning, the backup value $r + \gamma V(s')$ in Eq. 10 and Eq. 11 can be computed, similar to EPG, by integrating over all actions, i.e. $r + \gamma \sum_{a'} \pi(a'|s') Q(s', a')$. This is also consistent with the idea of Expected SARSA (Sutton et al., 1998).

We adopt the ideas of EPG and Expected SARSA to form our algorithm. Note that EPG can be applied in both on-policy and off-policy settings. Here we focus on off-policy actor-critic for improved sample efficiency. It is well known that target networks are essential in off-policy time-difference learning to mitigate non-stationarity and prevent divergence of the value function (Mnih et al., 2015). In our case, the backup value $r+\gamma \sum_{a'} \pi(a'|s') Q(s', a')$ depends on both π and Q, and thus requires target networks for both of them. Specifically, for the policy $\pi_\theta(\cdot|s)$ and value function $Q_\phi(s,a)$, we introduce corresponding target networks $\pi_{\bar\theta}(\cdot|s)$ and $Q_{\bar\phi}(s,a)$, where $\bar\theta$ and $\bar\phi$ are updated via slow tracking of θ and ϕ. In addition, when ρ is CVaR, we introduce an auxiliary function $q_\psi(s,a)$ to estimate quantile. Given a transition (s,a,r,s') sampled from the replay buffer, the policy $\pi_\theta(\cdot|s)$ is updated using EPG as in Eq. 9, while Q_ϕ (and q_ψ if applicable) is updated according to Eq. 10 or Eq. 11, where $V(s')$ is calculated using the target networks as $\sum_a \pi_{\bar\theta}(a|s) Q_{\bar\phi}(s,a)$. For continuous action spaces, the expectation in both policy gradient and value learning can be approximated in practice by sampling a sufficient number of actions. The full algorithm is in Algo. 1, with additional operations for dynamic CVaR marked in parenthesis.

Algorithm 1: Actor-Critic for dynamic Expectile (and CVaR)

Input: Risk level α, training steps M, mini batch size N, π learning rate η_θ, Q learning rate ζ_ϕ, (q learning rate ζ_ψ), soft update parameter τ;
Initialize: Policy π_θ, target policy $\pi_{\bar\theta}$, value function Q_ϕ, target value function $Q_{\bar\phi}$, (quantile value function q_ψ), buffer B;
$s =$ env.reset();
for $m = 1, \ldots, M$ **do**
  $a \sim \pi_\theta(\cdot|s)$; $r, s' \leftarrow$env.step(a) ;
  B.add(s, a, r, s');
  if B.*size()*$\geq N$ **then**
    Sample $\{(s,a,r,s')_i\}_{i=1}^N \sim B$;
    $V(s') = \mathbb{E}_{a'\sim\pi_{\bar\theta}(\cdot|s')}[Q_{\bar\phi}(s', a')]$;
    Update Q_ϕ (and q_ψ) via Eq. 10 or Eq. 11;
    Update π_θ via Eq. 9 with Q_ϕ using EPG;
    $\bar\theta \leftarrow \tau\theta + (1-\tau)\bar\theta$;
    $\bar\phi \leftarrow \tau\phi + (1-\tau)\bar\phi$;
  $s \leftarrow s'$;
  if *env.done()* **then**
    $s =$ env.reset();

3.4 Discussion on Other Morel-free Methods

Although the general policy gradient for dynamic risk requires transition perturbation (Eq. 8), several studies have attempted to develop model-free actor-critic methods for stochastic policies, which we review here. However, note that these methods are inherently on-policy and therefore cannot be directly extended to time-difference-style off-policy learning from replay buffers, as in our approach.

Jiang et al. (2024) proposed a mode-free, on-policy PG approach for dynamic expectile based on elicitability. Although the authors do not explicitly state that their algorithm optimizes dynamic expectile, we prove that it indeed does, see Appendix D. Analogous to Eq. 10, their Bellman operator for dynamic expectile is given by $\mathcal{T}^\pi V(s) = V(s) + 2\zeta \mathbb{E}_{a\sim\pi(\cdot|s)} \mathbb{E}_{r,s'} \big[(1-\alpha)\big(\delta(s,a,r,s')\big)_- +$

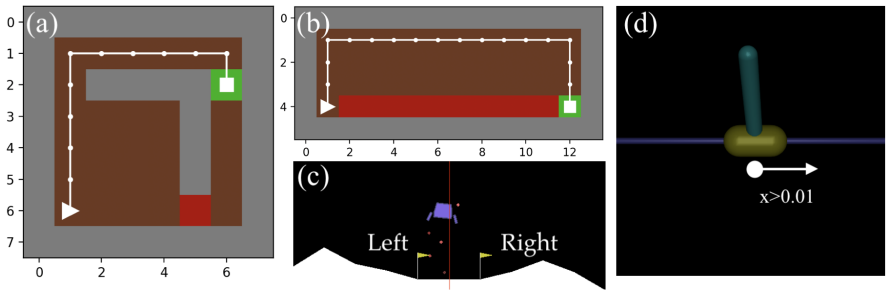


Figure 1: (a) Maze. Visiting red state receives a random reward, with mean -1. (b) Cliffwalk. Some states in row index 2 and 3 have chances to slide towards the cliff, see text. (c) LunarLander. Visiting all states in the right part receives a random reward with mean 0 per step. (d) Inverted Pendulum. Staying in the $x > 0.01$ region receives a random reward with mean 0 per step.

$\alpha\big(\delta(s,a,r,s')\big)_+\big]$, where $\delta(s,a,r,s') = r(s,a) + \gamma V(s') - V(s)$, and ζ is the step-size. Defining the advantage function as $A^\pi(s,a) := 2\zeta\mathbb{E}_{r,s'}\big[(1-\alpha)\big(\delta(s,a,r,s')\big)_- + \alpha\big(\delta(s,a,r,s')\big)_+\big]$, the policy is updated using the standard PG form $\mathbb{E}_s[A^\pi(s,a)\nabla_\theta \log \pi_\theta(a|s)]$. Their algorithm is instantiated within the PPO framework (Schulman et al., 2017).

Coache et al. (2023) developed an on-policy actor-critic method for dynamic CVaR. As discussed in Sec. 3.2, they learn the value function in a model-free manner by minimizing a jointly elicitable loss for the (VaR, CVaR) pair. The policy gradient, however, still requires solving for the optimal risk envelope via a Lagrangian formulation, i.e., $\max_\pi V(s) = \max_\pi \mathrm{CVaR}_\alpha(\tilde{r} + \gamma V(\tilde{s}')) = \max_\pi \min_\xi \max_\lambda \mathbb{E}[(\tilde{r} + \gamma V(\tilde{s}'))\xi(\tilde{r},\tilde{s}')] + \lambda(1 - \mathbb{E}[\xi(\tilde{r},\tilde{s}')])$. For CVaR, the optimal dual variable corresponds to the VaR (cf. Eq. 2), thus they replaced λ with the learned VaR function, denoted as $q(s)$. This yields an estimate of the risk envelope $\xi(r,s'|s,a) = \frac{1}{\alpha}\mathbb{I}\{r(s,a) + \gamma V(s') < q(s)\}$, and the resulting PG takes the form $\mathbb{E}_s\big[\frac{1}{\alpha}\big(r + \gamma V(s') - q(s)\big)_- \nabla_\theta \log \pi_\theta(a|s)\big]$ (see Eq. 6.16b and Eq. L2 in Coache et al. (2023)). Note that our formulation differs slightly, as we consider left-tail CVaR, whereas Coache et al. (2023) focus on the right tail.

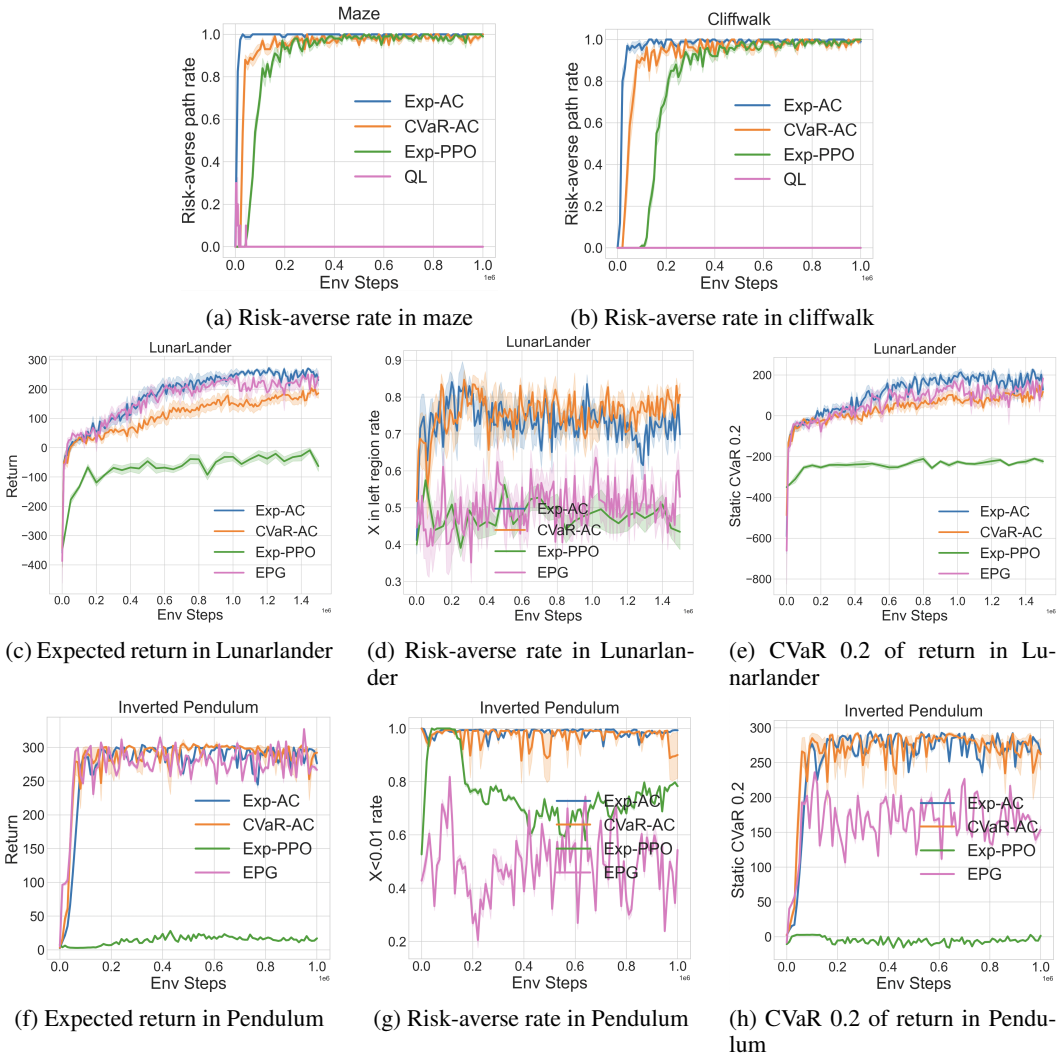
4 Experiments

Following Luo et al. (2024) and Mead et al. (2025), we modify several domains such that the risk-averse behavior is clear to identify to evaluate the algorithms. We start from two tabular cases: a maze adapted from Greenberg et al. (2022) and the Cliffwalk from Sutton et al. (1998), then followed with LunarLander from the Box2D environments of OpenAI Gym (Brockman et al., 2016), and InvertedPendulum from the Mujoco (Todorov et al., 2012) environments of OpenAI Gym. In the tabular domains, all algorithms are implemented in a tabular manner, whereas in the remaining environments, function approximation with neural networks is used. Since we are interested in learning risk-averse policies, we choose $\alpha < 0.5$ for expectile and $\alpha < 1$ for CVaR. Additional implementation, parameter details, and curves under different α are provided in Appendix E. We note that the modified Ant and HalfCheetah in Luo et al. (2024) are excluded because agents can move in both directions and we observe agents may achieve high returns while wandering near the origin, making risk-averse behavior difficult to characterize. Nevertheless, we instead evaluate our method on the HalfCheetah from Robust Gym (Gu et al., 2025) in Appendix E.3.3.

Baselines. We evaluate our proposed dynamic expectile (Exp-AC) and CVaR (CVaR-AC) actor-critics against model-free stochastic-policy baselines: Jiang et al. (2024) (denoted as Exp-PPO) and Coache et al. (2023). DPG methods (Sec. 2.2.1) are excluded due to discrete actions. However, due to under-performance and scale differences, learning curves for Coache et al. (2023) are reported separately in Appendix Fig. 3 to improve visualization. Q-learning (QL) (Sutton et al., 1998) and EPG (Ciosek & Whiteson, 2020) are included to serve as risk-neutral baselines.

4.1 Maze and Cliffwalk

The two domains are shown in Fig. 1 (a) and (b), where gray color marks the walls, and the agent starts from the white triangle and aims to reach the green goal. The action space is discrete with four actions {Up, Down, Left, Right}. For Maze, each step before reaching the goal gives a reward of -1, except at the red state, where the reward is $-1 + \mathcal{N}(0,1) \times 30$, clipped to $[-20, 20]$. Reaching the goal gives a reward of 10. Thus, the longer path (white) is risk-averse while the shortest path



(a) Risk-averse rate in maze

(b) Risk-averse rate in cliffwalk

(c) Expected return in Lunarlander

(d) Risk-averse rate in Lunarlander

(e) CVaR 0.2 of return in Lunarlander

(f) Expected return in Pendulum

(g) Risk-averse rate in Pendulum

(h) CVaR 0.2 of return in Pendulum

Figure 2: Risk-averse path rate in Maze and Cliffwalk. Expected return, risk-averse rate, and CVaR 0.2 of return in LunarLander and Inverted Pendulum. Curves are averaged over 10 seeds with shaded regions indicating standard errors.

going through the red state is risk-neutral. For Cliffwalk, each safe step gives -1, while entering a red cliff state gives -100 and terminates the episode. For states directly above the cliff (row 3, columns $[2, 11]$), left/right actions have a 20% chance of sliding into the cliff; for states in row 2, columns $[2, 7]$, the corresponding probability of sliding to row 3 is 10%. This makes the middle path risk-neutral and the longer white path risk-averse. Since tabular learning is relatively stable, we use small risk levels, e.g., $\alpha = 0.05$ for Exp-AC and Exp-PPO, and $\alpha = 0.1$ for CVaR-AC.

Agents are evaluated by sampling 10 trajectories during training and we report the risk-averse path rate of sampled trajectories, as shown in Fig. 2a and 2b. In both domains, the three risk-averse algorithms can successfully identify the risk-averse path, with Exp-AC and CVaR-AC learning faster than Exp-PPO, since off-policy learning is more sample efficient than on-policy learning.

4.2 LunarLander

This domain is illustrated in Fig. 1(c), where the objective is to control the lander to achieve a safe landing. We use the discrete-action version of the environment and refer readers to the official documentation for details. As shown in the figure, the landing pad is divided into left and right regions by its center line. In Luo et al. (2024); Mead et al. (2025), random rewards are introduced only when the agent lands in the right region. Since we consider dynamic risk, we instead inject noise at every time step. Specifically, whenever the lander is in the right region, either during flight or upon landing, we add a noisy reward $\mathcal{N}(0, 1) \times 3$, clipped to $[-6, 6]$. Consequently, a risk-

averse agent is expected to avoid the right region while still achieving a safe landing. The maximum episode length is 500. $\alpha = 0.3 \in \{0.2, 0.3, 0.4\}$ for Exp-AC. $\alpha = 0.9 \in \{0.5, 0.7, 0.9\}$ for CVaR-AC. $\alpha = 0.4 \in \{0.2, 0.3, 0.4\}$ for Exp-PPO.

The return and the rate of staying in non-noisy (left) region in an episode are shown in Fig. 2c and 2d. Since dynamic risk is difficult to measure directly in practice, we additionally report the empirical 0.2-CVaR of the return in Fig. 2e as a reference. This is estimated by sampling 15 episodes during evaluation and averaging the three worst returns. Exp-AC achieves both a higher risk-averse rate and a higher 0.2-CVaR compared to the risk-neutral EPG baseline. It also achieves a high left landing rate as shown in Fig. 6 in appendix. CVaR-AC is more challenging to tune in this domain and exhibits slightly lower performance than EPG, although it still demonstrates a clear risk-averse tendency (cf. Fig. 2d). In contrast, Exp-PPO fails to learn a reasonable policy in this setting.

4.3 Inverted Pendulum

This domain is illustrated in Fig. 1 (d). The objective is to keep the inverted pendulum upright (within a certain angle limit) as long as possible. The original per-step reward is 1. We add a noise $\mathcal{N}(0, 1) \times 10$, clipped to $[-20, 20]$, to the per-step reward if X-position> 0.01. Thus, a risk-averse agent is expected to stay away from the noisy reward region while keep the balance of the pendulum. The maximum episode length is 300. $\alpha = 0.2 \in \{0.2, 0.3, 0.4\}$ for Exp-AC. $\alpha = 0.7 \in \{0.5, 0.7, 0.9\}$ for CVaR-AC. $\alpha = 0.4 \in \{0.2, 0.3, 0.4\}$ for Exp-PPO.

The return, $X < 0.01$ rate in an episode, and 0.2-CVaR of return are presented in Fig. 2f, 2g, and 2h. Both Exp-AC and CVaR-AC learn risk-averse policies by holding the pendulum more in the non-noisy region compared with EPG. Exp-PPO fails to learn a reasonable policy within the same environment step. Curves with more steps for Exp-PPO are shown in Fig. 8 in appendix, which exhibits unstable policy learning.

Discussion. We observe that the method in Coache et al. (2023) generally performs poorly across the considered domains under the same hyperparameter search range. One possible explanation is that its policy gradient depends on the optimal dual variable and transition perturbation, both of which are approximated using learned value functions and may therefore introduce additional estimation error. For Exp-PPO (Jiang et al., 2024), we report results using single-step advantage estimation. Its nonlinear advantage function (cf. Sec. 3.4) makes it difficult to construct GAE (Schulman et al., 2015) as in risk-neutral PPO. Although the authors propose a multi-step alternative based on repeated Bellman updates, we find that it leads to degraded performance. CVaR-AC jointly learns a policy and two value functions, resulting in a three-timescale algorithm. This additional complexity makes hyperparameter tuning more challenging, as observed in LunarLander. We also find that both Exp-AC and CVaR-AC exhibit degraded performance in LunarLander when increasing risk-aversion (Fig. 5). One possible explanation is the presence of epistemic uncertainty during learning: overly strong risk aversion may discourage exploration and consequently slow policy improvement.

5 Conclusion and Future Work

This paper proposes a model-free off-policy actor–critic algorithm with stochastic policies for dynamic expectile and CVaR under softmax policy parameterization. The main advantages are that it avoids transition perturbation required by most stochastic-policy methods and improves sample efficiency via off-policy learning. We evaluate the method on both discrete and continuous action domains, under tabular settings and function approximation. Empirically, we show that our method succeeds in settings where existing approaches fail to learn a risk-averse or a reasonable policy.

Limitations and future work. The value learning methods in Sec. 3.2 are specific to expectile and CVaR and are not directly applicable to other risk measures. Moreover, optimizing dynamic CVaR leads to a three-timescale update scheme, which increases the difficulty of hyperparameter tuning. Observing the two limitations, extending the framework to other risk measures and improving the stability of dynamic CVaR training remain valuable for future work.

AI use statement

In this work, we only used generative AI tools for sentence refinement.

Reproducibility statement

Theoretical part: for proving the theorems and propositions, the general assumptions we made are the random variable are continuous and the quantile is unique, as stated in Sec. 2.1, and the assumptions listed in assumption 3.1. An additional assumption for proving Proposition 3.5 is listed when stating the proposition.

Empirical part: the codebase is included in the supplementary material. The parameter settings are discussed in Appendix E.

References

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Fabio Bellini and Valeria Bignozzi. On elicitable risk measures. *Quantitative Finance*, 15(5):725–733, 2015.

Fabio Bellini and Elena Di Bernardino. Risk management with expectiles. *The European Journal of Finance*, 23(6):487–506, 2017.

Axel Benyamine, Julien Grand-Clément, Marek Petrik, Michael I Jordan, and Alain Durmus. Dynamic programming for epistemic uncertainty in markov decision processes. *arXiv preprint arXiv:2602.03381*, 2026.

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*, volume 1. Athena Scientific, 1996.

Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint, Second Edition*, volume 100. Springer, 2023.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and trends in Machine Learning*, 8(3-4):231–357, 2015.

Kamil Ciosek and Shimon Whiteson. Expected policy gradients for reinforcement learning. *Journal of Machine Learning Research*, 21(52):1–51, 2020.

Anthony Coache, Sebastian Jaimungal, and Alvaro Cartea. Conditionally elicitable dynamic risk measures for deep reinforcement learning. *SIAM Journal on Financial Mathematics*, 14(4):1249–1289, 2023.

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, volume 32, 2018.

Tobias Fissler and Johanna F Ziegel. Higher order elicitability and osband's principle. *The Annals of Statistics*, pp. 1680–1707, 2016.

Ido Greenberg, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Efficient risk-averse reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:32639–32652, 2022.

Shangding Gu, Laixi Shi, Muning Wen, Ming Jin, Eric Mazumdar, Yuejie Chi, Adam Wierman, and Costas Spanos. Robust gymnasium: A unified modular benchmark for robust reinforcement learning. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Audrey Huang, Liu Leqi, Zachary C Lipton, and Kamyar Azizzadenesheli. On the convergence and optimality of policy gradient for markov coherent risk. *arXiv preprint arXiv:2103.02827*, 2021.

Yuhua Jiang, Qihan Liu, Xiaoteng Ma, Chenghao Li, Yiqin Yang, Jun Yang, Bin Liang, and Qianchuan Zhao. Learning diverse risk preferences in population-based self-play. In *Proceedings of the aaai conference on artificial intelligence*, volume 38, pp. 12910–12918, 2024.

Shiau Hong Lim and Ilyas Malik. Distributional reinforcement learning for risk-sensitive policies. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:30977–30989, 2022.

Yudong Luo, Yangchen Pan, Han Wang, Philip Torr, and Pascal Poupart. A simple mixture policy parameterization for improving sample efficiency of cvar optimization. *Reinforcement Learning Journal*, 2024.

Saeed Marzban, Erick Delage, and Jonathan Yu-Meng Li. Deep reinforcement learning for option pricing and hedging under dynamic expectile risk measures. *Quantitative finance*, 23(10):1411–1430, 2023.

Harry Mead, Clarissa Costen, Bruno Lacerda, and Nick Hawes. Return capping: Sample-efficient cvar policy gradient optimisation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Marc Rigter, Bruno Lacerda, and Nick Hawes. One risk to rule them all: A risk-sensitive perspective on model-based offline reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:77520–77545, 2023.

R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Andrzej Ruszczyński. Risk-averse dynamic programming for markov decision processes. *Mathematical programming*, 125(2):235–261, 2010.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

Yun Shen, Wilhelm Stannat, and Klaus Obermayer. Risk-sensitive markov control processes. *SIAM Journal on Control and Optimization*, 51(5):3652–3672, 2013.

Yun Shen, Michael J Tobia, Tobias Sommer, and Klaus Obermayer. Risk-sensitive reinforcement learning. *Neural computation*, 26(7):1298–1328, 2014.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 387–395. Pmlr, 2014.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Sequential decision making with coherent risk. *IEEE transactions on automatic control*, 62(7):3323–3338, 2016.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Runyu Zhang, Yang Hu, and Na Li. Soft robust MDPs and risk-sensitive MDPs: Equivalence, policy gradient, and sample complexity. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

A Derivations

A.1 SPG for Coherent Dynamic Risk

The objective we consider for dynamic risk with $\pi(\cdot|s)$ is

$$\rho^{\pi}_{\infty}(M) = \sum_{a_0} \pi(a_0|s_0)\rho\Big(\tilde{r}(s_0,a_0)+\gamma \sum_{a_1} \pi(a_1|\tilde{s}_1)\rho\big(\tilde{r}(\tilde{s}_1,a_1)+\gamma \sum_{a_2} \pi(a_2|\tilde{s}_2)\rho(\tilde{r}(\tilde{s_2},a_2)+...)\big)\Big)$$

Similar to deterministic policy case, the action-value function and state value function are defined as $Q^{\pi}(s,a) = \rho^{\pi}_{\infty}(M|s_0 = s, a_0 = a)$ and $V^{\pi}(s) = \rho^{\pi}_{\infty}(M|s_0 = s)$. It holds that $V^{\pi}(s) = \sum_a \pi(a|s)Q^{\pi}(s,a)$. The gradient of policy parameter θ w.r.t. $V^{\pi}(s_0)$ is computed as follows.

$$\begin{aligned}
\nabla_\theta V^{\pi_\theta}(s_0) &= \nabla_\theta \sum_{a_0} \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0,a_0) \\
&= \sum_{a_0} \nabla_\theta \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0,a_0) + \pi_\theta(a_0|s_0)\nabla_\theta Q^{\pi_\theta}(s_0,a_0) \\
&\overset{(a)}{=} \mathbb{E}_{a_0\sim\pi_\theta(\cdot|s_0)}\Big[Q^{\pi_\theta}(s_0,a_0)\nabla_\theta \log \pi_\theta(a_0|s_0) + \nabla_\theta Q^{\pi_\theta}(s_0,a_0)\Big] \\
&\overset{(b)}{=} \mathbb{E}_{a_0\sim\pi_\theta(\cdot|s_0)}\Big[Q^{\pi_\theta}(s_0,a_0)\nabla_\theta \log \pi_\theta(a_0|s_0) + \nabla_\theta \rho(\tilde{r}_0 + \gamma V^{\pi_\theta}(\tilde{s}_1))\Big] \\
&\overset{(c)}{=} \mathbb{E}_{a_0\sim\pi_\theta(\cdot|s_0)}\Big[Q^{\pi_\theta}(s_0,a_0)\nabla_\theta \log \pi_\theta(a_0|s_0) + \nabla_\theta \mathbb{E}_{r_0,s_1\sim P^{\xi^*}(\cdot,\cdot|s_0,a_0)}\big[r_0 + \gamma V^{\pi_\theta}(s_1)\big]\Big] \\
&= \mathbb{E}_{a_0\sim\pi_\theta(\cdot|s_0)}\Big[Q^{\pi_\theta}(s_0,a_0)\nabla_\theta \log \pi_\theta(a_0|s_0) + \gamma \mathbb{E}_{r_0,s_1\sim P^{\xi^*}(\cdot,\cdot|s_0,a_0)}\nabla_\theta V^{\pi_\theta}(s_1)\Big],
\end{aligned}$$

where (a) uses $\nabla_\theta \pi_\theta(a|s) = \pi_\theta(a|s)\nabla_\theta \log \pi(a|s)$; (b) follows the Bellman equation of $Q^{\pi}(s,a)$; (c) uses the dual representation of $\rho(\cdot)$ with the optimal risk envelope P^{ξ^*}.

Calculating $\nabla_\theta V^{\pi_\theta}(s_1)$ follows the same step as calculating $\nabla_\theta V^{\pi_\theta}(s_0)$. Putting all gradients back yields Eq. 8.

A.2 Surrogate SPG with Softmax Policy

When $\pi(\cdot|s)$ is a softmax policy, $\pi_\theta(a|s) = \exp(\theta(s,a))/\sum_b \exp(\theta(s,b))$. Therefore

$$\frac{\partial \log \pi_\theta(a|s)}{\partial \theta(s,i)} = \mathbb{I}\{a=i\} - \pi_\theta(i|s)$$

For each s, the gradient to compute in Eq. 9 is $\nabla_\theta d(s) \sum_a \pi_\theta(a|s)Q^{\pi}(s,a)$. Consider the partial derivative for $\theta(s,i)$

$$\begin{aligned}
\frac{\partial \sum_a \pi_\theta(a|s)Q^{\pi}(s,a)}{\partial \theta(s,i)} &= \sum_a Q^{\pi}(s,a)\frac{\partial \pi_\theta(a|s)}{\partial \theta(s,i)} = \sum_a Q^{\pi}(s,a)\pi_\theta(a|s)\frac{\partial \log \pi_\theta(a|s)}{\partial \theta(s,i)} \\
&= \sum_a Q^{\pi}(s,a)\pi_\theta(a|s)\big(\mathbb{I}\{a=i\} - \pi_\theta(i|s)\big) \\
&= \sum_a Q^{\pi}(s,a)\pi_\theta(a|s)\mathbb{I}\{a=i\} - \sum_a Q^{\pi}(s,a)\pi_\theta(a|s)\pi_\theta(i|s) \\
&= Q^{\pi}(s,i)\pi_\theta(i|s) - \pi_\theta(i|s)V^{\pi}(s) \\
&= \pi_\theta(i|s)\big(Q^{\pi}(s,a) - V^{\pi}(s)\big)
\end{aligned}$$

Therefore, $\forall (s,a) \in \mathcal{S}\times\mathcal{A}$, $\theta(s,a)$ is updated by $\theta(s,a) \leftarrow \theta(s,a) + \eta\, d(s)\pi_\theta(a|s)(Q^{\pi}(s,a) - V^{\pi}(s))$. This calculation is the same as in the risk-neutral setting.

A.3 JOINT ELICITABILITY OF CVAR

Corollary 5.5 of Fissler & Ziegel (2016) gives a general form of loss function for $(\mathrm{VaR}_\alpha, \mathrm{CVaR}_\alpha)$,

$$\mathrm{VaR}_\alpha(\tilde{x}), \mathrm{CVaR}_\alpha(\tilde{x}) = \arg\min_{y_1,y_2} \mathbb{E}[L(\tilde{x}, y_1, y_2)], \text{ with}$$

$$L(x, y_1, y_2) = \Big(\mathbb{I}_{\{x\le y_1\}} - \alpha\Big)G_1(y_1) - \mathbb{I}_{\{x\le y_1\}}G_1(x) + G_2(y_2)\Big(y_2 - y_1 + \frac{1}{\alpha}\mathbb{I}_{\{x\le y_1\}}(y_1 - x)\Big) - G_3(y_2) + \phi(x),$$

where G_1 is increasing, $G_3' = G_2$, G_3 is strictly increasing and strictly convex. The partial derivatives of L are

$$\frac{\partial L}{\partial y_1} = (\mathbb{I}\{x \le y_1\} - \alpha)\big(G_1'(y_1) + \frac{G_2(y_2)}{\alpha}\big)$$
$$\frac{\partial L}{\partial y_2} = G_2'(y_2)\big(y_2 - y_1 + \frac{1}{\alpha}\mathbb{I}\{x \le y_1\}(y_1 - x)\big).$$

Note that in $\frac{\partial L}{\partial y_1}$, the $\mathbb{I}\{x \le y_1\} - \alpha$ term is the gradient of quantile regression. But it also contains a term that depends on the value of y_2 (which is the CVaR estimator). In $\frac{\partial L}{\partial y_2}$, $\mathbb{E}[y_2 - y_1 + \frac{1}{\alpha}\mathbb{I}\{\tilde{x} \le y_1\}(y_1 - \tilde{x})] = y_2 - \mathrm{CVaR}_\alpha(\tilde{x})$ if $y_1 = \mathrm{VaR}_\alpha(\tilde{x})$, which directly quantifies the difference between y_2 and $\mathrm{CVaR}_\alpha(\tilde{x})$, but it also contains a term $G_2'(y_2)$. This motivates us to investigate if those additional terms can be removed so as to improve the stability of gradient updates.

In practice, quantile is elicitable via quantile regression loss, and α-CVaR can be written as an expectation if α-quantile is known (cf. Eq. 2), i.e.,

$$\mathrm{VaR}_\alpha(\tilde{x}) = \arg\min_{y_1} \mathbb{E}[l_\alpha(\tilde{x} - y_1)], \;\; l_\alpha(x - y) := (\alpha - \mathbb{I}\{x < y\})(x - y)$$
$$\mathrm{CVaR}_\alpha(\tilde{x}) = \arg\min_{y_2} \frac{1}{2}\Big(y_2 - \big(\mathrm{VaR}_\alpha - \frac{1}{\alpha}\mathbb{E}[(\mathrm{VaR}_\alpha - \tilde{x})_+]\big)\Big)^2.$$

This suggests to update y_1 and y_2 via

$$y_1 \leftarrow y_1 - \zeta_1 \cdot \partial_{y_1}\mathbb{E}[l_\alpha(\tilde{x} - y_1)] = y_1 - \zeta_1 \cdot \mathbb{E}[\mathbb{I}\{\tilde{x} < y_1\} - \alpha]$$
$$y_2 \leftarrow y_2 - \zeta_2 \cdot \big(y_2 - (y_1 - \frac{1}{\alpha}\mathbb{E}[(y_1 - \tilde{x})_+])\big).$$

When using a single sample x of $\tilde{x}$, the update rules become

$$y_1 \leftarrow y_1 - \zeta_1 \cdot (\mathbb{I}\{\tilde{x} < y_1\} - \alpha)$$
$$y_2 \leftarrow y_2 - \zeta_2 \cdot \big(y_2 - y_1 + \frac{1}{\alpha}(y_1 - \tilde{x})_+\big).$$

Comparing with the gradient of the joint elicitable loss function, the terms $G_1'(y_1) + G_2(y_2)/\alpha$ and $G_2'(y_2)$ are removed. We will show in Proposition 3.5 that proper convergence guarantees of this learning procedure can be obtained using two-time-scale stochastic optimization arguments.

B PROOFS FOR SECTION 3.1

B.1 PROOF OF THEOREM 3.2

We first recall a Lemma from Agarwal et al. (2021).

Lemma B.1. *[Lemma 52 of Agarwal et al. (2021)] Fix a state s. Let $\theta_s \in \mathbb{R}^{|\mathcal{A}|}$ be the column vector of parameters for state s. Let $\pi_\theta(\cdot|s)$ be the corresponding vector of action probabilities given by the softmax parameterization. For some fixed vector $c \in \mathbb{R}^{|\mathcal{A}|}$, define:*

$$F(\theta) := \sum_a \pi_\theta(a|s)c_a.$$

Then $||\nabla_{\theta_s}F(\theta_s) - \nabla_{\theta_s}F(\theta_s')||_2 \le \beta||\theta_s - \theta_s'||_2$ where $\beta = 5||c||_\infty$.

To prove Thorem 3.2, we first prove the following lemma.

Lemma B.2. *With assumption 3.1 (specifically, $\tilde{r} \in [0,1]$ and $d(s) \in (0,1]$). Let π_θ be a softmax policy. For update rule in Eq. 9, if $\eta \leq (1-\gamma)/5$, then $\mathbb{E}_{a\sim\pi_{t+1}(\cdot|s)}[Q^{\pi_t}(s,a)] \geq \mathbb{E}_{a\sim\pi_t(\cdot|s)}[Q^{\pi_t}(s,a)]$ for all $s \in \mathcal{S}$.*

Proof. Let $F_s^Q(\theta) := d(s)\sum_a \pi_\theta(a|s)Q(s,a)$. By Lemma B.1, we have

$$\begin{aligned}
\|\nabla_{\theta_s} F_s^Q(\theta) - \nabla_{\theta'_s} F_s^Q(\theta')\|_2 &= \|d(s)(\nabla_{\theta_s}\sum_a \pi_\theta(a|s)Q(s,a) - \nabla_{\theta'_s}\sum_a \pi_{\theta'}(a|s)Q(s,a))\|_2 \\
&= d(s)\|\nabla_{\theta_s}\sum_a \pi_\theta(a|s)Q(s,a) - \nabla_{\theta'_s}\sum_a \pi_{\theta'}(a|s)Q(s,a)\|_2 \\
&\overset{(a)}{\leq} \|\nabla_{\theta_s}\sum_a \pi_\theta(a|s)Q(s,a) - \nabla_{\theta'_s}\sum_a \pi_{\theta'}(a|s)Q(s,a)\|_2 \\
&\overset{(b)}{\leq} 5\|Q\|_\infty \|\theta(s,\cdot) - \theta'(s,\cdot)\|_2 \\
&\overset{(c)}{=} 5/(1-\gamma)\|\theta(s,\cdot) - \theta'(s,\cdot)\|_2,
\end{aligned}$$

where (a) follows $d(s) \leq 1$; (b) follows Lemma B.1; (c) is due to the assumption $\tilde{r} \in [0,1]$, then $||Q||_\infty \leq \frac{1}{1-\gamma}$.

Hence, $F_s^Q(\theta)$ is $\beta := 5/(1-\gamma)$ smooth with respect to θ_s. We further observe that the policy update can be rewritten as: $\theta_{t+1}(s,\cdot) := \theta_t(s,\cdot) + \eta\nabla_{\theta_s} F_s^{Q^{\pi_t}}(\theta_t),\ \forall s$. Recall that for a β smooth function, gradient ascent will increase the function value provided that $\eta \leq 1/\beta$ (Beck, 2017). So we get that if $\beta \leq (1-\gamma)/5$, then $F_s^{Q^{\pi_t}}(\theta_{t+1}) \geq F_s^{Q^{\pi_t}}(\theta_t)$, which means $\sum_a \pi_{t+1}(a|s)Q^{\pi_t}(s,a) \geq \sum_a \pi_t(a|s)Q^{\pi_t}(s,a), \forall s$. □

Next, we prove Theorem 3.2.

Theorem 3.2. *With assumption 3.1, for softmax policy, the update rule in Eq. 9 with $\eta \leq (1-\gamma)/5$ ensures that for all $(s,a) \in \mathcal{S}\times\mathcal{A}$, $Q^{\pi_{t+1}}(s,a) \geq Q^{\pi_t}(s,a),\ V^{\pi_{t+1}}(s) \geq V^{\pi_t}(s)$.*

Proof. From Lemma B.2, we have $\mathbb{E}_{a\sim\pi_{t+1}(\cdot|s)}[Q^{\pi_t}(s,a)] \geq \mathbb{E}_{a\sim\pi_t(\cdot|s)}[Q^{\pi_t}(s,a)], \forall s$. Recall the Bellman operator in Eq. 7

$$\mathcal{T}^\pi Q(s,a) = \rho\big(\tilde{r}(s,a) + \gamma\mathbb{E}_{a'\sim\pi(\cdot|\tilde{s}')}[Q(\tilde{s}',a')]\big)$$

By the monotonicity of coherent risk measure and Lemma B.2, we have

$$\rho\big(r(s,a) + \gamma\mathbb{E}_{a'\sim\pi_{t+1}(\cdot|\tilde{s}')}[Q^{\pi_t}(\tilde{s}',a')]\big) \geq \rho\big(r(s,a) + \gamma\mathbb{E}_{a'\sim\pi_t(\cdot|\tilde{s}')}[Q^{\pi_t}(\tilde{s}',a')]\big),$$

which means $\mathcal{T}^{\pi_{t+1}}Q^{\pi_t} \geq \mathcal{T}^{\pi_t}Q^{\pi_t} = Q^{\pi_t}$.

Futhermore, by the monotonicity of coherent risk measure, we have

$$Q^{\pi_t} \leq \mathcal{T}^{\pi_{t+1}}Q^{\pi_t} \leq (\mathcal{T}^{\pi_{t+1}})^2 Q^{\pi_t} \leq ... \leq \lim_{k\to\infty}(\mathcal{T}^{\pi_{t+1}})^k Q^{\pi_t} = Q^{\pi_{t+1}}.$$

The monotone increasing of V is due to

$$\begin{aligned}
&V^{\pi_{t+1}}(s) - V^{\pi_t}(s) \\
=&\sum_a \pi_{t+1}(a|s)[Q^{\pi_{t+1}}(s,a) - Q^{\pi_t}(s,a)] + \sum_a [\pi_{t+1}(a|s) - \pi_t(a|s)]Q^{\pi_t}(s,a)
\end{aligned}$$

The first term is non-negative since we have shown $Q^{\pi_{t+1}} \geq Q^{\pi_t}$. The second term is also non-negative from Lemma B.2. Thus $V^{\pi_{t+1}} \geq V^{\pi_t}$. □

B.2 Proof of Theorem 3.3

Our proof adapts the proof of Theorem 10 in Agarwal et al. (2021). One can start with a lemma that confirms the convergence of the iterates.

Lemma B.3 (Lemma 42 in Agarwal et al. (2021)). *For all states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$, the sequences $\{V^{\pi_t}(s)\}_{t\geq 0}$ and $\{Q^{\pi_t}(s,a)\}_{t\geq 0}$ are non-decreasing and converge:*

$$V^{\pi_t}(s) \to V^{\infty}(s), \qquad Q^{\pi_t}(s,a) \to Q^{\infty}(s,a).$$

Define the limiting advantage function $A^{\infty}(s,a) = Q^{\infty}(s,a) - V^{\infty}(s)$, and let $\Delta := \min_{(s,a):\, A^{\infty}(s,a)>0} A^{\infty}(s,a)$. Then $\Delta > 0$. Moreover, there exists a finite iteration index T_0 such that for all $t > T_0$, all $s \in \mathcal{S}$, and all $a \in \mathcal{A}$, $Q^{\pi_t}(s,a) > Q^{\infty}(s,a) - \Delta/4$.

The existence of Q^{∞} and V^{∞} is ensured by the monotone increasing of the value function (Theorem 3.2) and that value functions are bounded above by $1/(1-\gamma)$.

The proof relies on defining for each state s, a partition of the actions into three sets:

$$\begin{aligned} I_s^0 &:= \{a | Q^{\infty}(s,a) = V^{\infty}(s)\}, \\ I_s^+ &:= \{a | Q^{\infty}(s,a) > V^{\infty}(s)\}, \\ I_s^- &:= \{a | Q^{\infty}(s,a) < V^{\infty}(s)\}, \end{aligned}$$

and showing in Part 1 that $\pi_t(a|s) \to 0$ for $a \in I_s^+ \cup I_s^-$ followed with $I_s^+ = \emptyset$ in Part 2. With the latter property in hand, one can show that

$$\begin{aligned} \rho\big(\tilde{r}(s,a) + \gamma \cdot \max_{a'} Q^{\infty}(\tilde{s}', a)\big) &= \rho\big(\tilde{r}(s,a) + \gamma \cdot \max_{a' \in I_s^0 \cup I_s^-} Q^{\infty}(\tilde{s}', a)\big) \\ &\leq \rho\big(\tilde{r}(s,a) + \gamma \cdot \max_{a' \in I_s^0 \cup I_s^-} V^{\infty}(\tilde{s}')\big) \\ &= \rho\big(\tilde{r}(s,a) + \gamma V^{\infty}(\tilde{s}')\big) = Q^{\infty}(s,a), \end{aligned}$$

where we first exploit the fact that I_s^+ is empty and where the final equality follows form the fact that $Q^{\pi_t}(s,a) = \rho\big(\tilde{r}(s,a) + \gamma V^{\pi_t}(\tilde{s}')\big)$ for all t, hence must be satisfied in the limit. This can be combined to

$$\begin{aligned} \rho\big(\tilde{r}(s,a) + \gamma \cdot \max_{a'} Q^{\pi_t}(\tilde{s}', a)\big) &\geq \rho\big(\tilde{r}(s,a) + \gamma \cdot \sum_{a'} \pi_t(a'|\tilde{s}') Q^{\pi_t}(\tilde{s}', a')\big) \\ &= \rho\big(\tilde{r}(s,a) + \gamma V^{\pi_t}(\tilde{s}')\big) = Q^{\pi_t}(s,a), \end{aligned}$$

and therefore in the limit to obtain that Q^{∞} satisfies Eq. 5 and thus both $Q^{\infty} = Q^*$ and (by construction) $V^{\infty} = V^*$. This completes the proof of Theorem 3.3.

B.2.1 Part 1

We first show that $\pi_t(a|s) \to 0$ for $a \in I_s^+ \cup I_s^-$ as $t \to \infty$. The proof of this part differs from that in Agarwal et al. (2021). In particular, Agarwal et al. (2021) establishes the vanishing of the policy gradient by exploiting the smoothness of the associated antiderivative (see Lemma 44 therein), whereas our surrogate policy gradient does not admit a corresponding antiderivative with such structure.

Define $J(\theta) := \sum_s d(s) \sum_a \pi_\theta(a|s) Q^{\pi_\theta}(s,a)$. Define the local primitive function as $\mathcal{L}_t(\theta) := \sum_s d(s) \sum_a \pi_\theta(a|s) Q^{\pi_{\theta_t}}(s,a)$. As a result, the surrogate policy gradient (Eq. 9) can be expressed as $g_t(\theta_t) := \nabla_\theta \mathcal{L}_t(\theta)|_{\theta=\theta_t}$.

Lemma B.4. *With the same conditions in Theorem 3.2, $g_t(\theta_t) \to 0$ as $t \to \infty$.*

Proof. From Theorem 3.2,

$$\begin{aligned} & J(\theta_{t+1}) - \mathcal{L}_t(\theta_{t+1}) \\ =& \sum_s d(s) \sum_a \pi_{\theta_{t+1}}(a|s) Q^{\pi_{\theta_{t+1}}}(s,a) - \sum_s d(s) \sum_a \pi_{\theta_{t+1}}(a|s) Q^t(s,a) \\ =& \sum_s d(s) \sum_a \pi_{\theta_{t+1}}(a|s) [Q^{\pi_{\theta_{t+1}}}(s,a) - Q^t(s,a)] \geq 0 \end{aligned}$$

Also, from Lemma B.1 with and the proof of Lemma B.2, we have

$$\|\nabla_\theta \mathcal{L}_t(\theta) - \nabla_\theta \mathcal{L}_t(\theta')\|_2^2 = \sum_s ||\nabla_{\theta_s} F_s^Q(\theta_s) - \nabla_{\theta_s} F_s^Q(\theta_s')||_2^2 \leq \beta^2 \sum_s ||\theta_s - \theta_s'||_2^2 \leq \beta^2 ||\theta - \theta'||_2^2,$$

hence $\mathcal{L}_t(\theta)$ is $\beta := 5/(1-\gamma)$ smooth. For a β-smooth function f, it holds that (see Lemma 3.4 of Bubeck (2015))

$$|f(\theta_{t+1}) - f(\theta_t) - \nabla_\theta f(\theta_t)^\top(\theta_{t+1} - \theta_t)| \leq \frac{\beta}{2}\|\theta_{t+1} - \theta_t\|^2.$$

When $\theta_{t+1} = \theta_t + \eta\nabla_\theta f(\theta_t)$,

$$f(\theta_{t+1}) \geq f(\theta_t) + \eta\|\nabla_\theta f(\theta_t)\|^2 - \frac{\beta}{2}\|\eta\nabla_\theta f(\theta_t)\|^2,$$

or equivalently,

$$f(\theta_{t+1}) - f(\theta_t) \geq \eta(1 - \frac{\beta\eta}{2})\|\nabla_\theta f(\theta_t)\|^2.$$

If learning rate $\eta \leq \frac{1}{\beta}$, $f(\theta_{t+1}) - f(\theta_t) \geq \frac{\eta}{2}\|\nabla_\theta f(\theta_t)\|^2$. Therefore,

$$\mathcal{L}_t(\theta_{t+1}) - \mathcal{L}_t(\theta_t) \geq \frac{\eta}{2}\|g_t(\theta_t)\|^2.$$

As a result,

$$J(\theta_{t+1}) - J(\theta_t) = J(\theta_{t+1}) - \mathcal{L}_t(\theta_t) \geq \mathcal{L}_t(\theta_{t+1}) - \mathcal{L}_t(\theta_t) \geq \frac{\eta}{2}\|g_t(\theta_t)\|^2 \geq 0.$$

Since $J(\theta_t)$ is non-decreasing in t and $J(\theta)$ is bounded above by $1/(1-\gamma)$, it must hold that

$$\sum_{t=0}^{T-1} \frac{\eta}{2}\|g_t(\theta_t)\|^2 \leq J(\theta_T) - J(\theta_0) \leq \frac{1}{1-\gamma} - J(\theta_0) < \infty,\ \forall T$$

which means $\lim_{t\to\infty} \|g_t(\theta_t)\|^2 = 0$, and therefore $\lim_{t\to\infty} g_t(\theta_t) = 0$. □

From Appendix A.2, we know

$$[g_t(\theta_t)](s,a) = d(s)\pi_{\theta_t}(a|s)A^t(s,a).$$

The following lemma states the magnitude of $A^t(s,a)$ for $a \in I_s^+ \cup I_s^-$.

Lemma B.5. *[Lemma 43 of Agarwal et al. (2021)] Define $A^t(s,a) := Q^t(s,a) - V^t(s)$. There exists a T_1 such that for all $t > T_1$, for all states s and actions a, we have*

$$A^t(s,a) < -\frac{\Delta}{4},\ \forall a \in I_s^-; \quad A^t(s,a) > \frac{\Delta}{4},\ \forall a \in I_s^+; \text{ where}$$
$$\Delta = \min_{\{s,a|A^\infty(s,a)\neq 0\}} |A^\infty(s,a)|, \text{ and } A^\infty(s,a) = Q^\infty(s,a) - V^\infty(s)$$

The proof of Lemma B.5 follows the same steps as in Agarwal et al. (2021).

Since when $t > T_1$, $|A^t(s,a)| > \Delta/4$ for $a \in I_s^+ \cup I_s^-$ (Lemma B.5), we must also have that

$$|[g_t(\theta_t)](s,a)| = d(s)\pi_{\theta_t}(s,a)|A^t(s,a)| \geq (\min_{s'} d(s'))\frac{\Delta}{4}\pi_{\theta_t}(s,a).$$

Hence, $[g_t(\theta_t)](s,a) \to 0$ as $t \to \infty$ (Lemma B.4), which implies that $\pi_t(a|s) \to 0$ for $a \in I_s^+ \cup I_s^-$. This completes the proof of this part.

B.2.2 PART 2

In this part, we show that $I_s^+ = \emptyset$. This is proved by contradiction and follows the same steps as in Agarwal et al. (2021). To shorten our discussion we omit to present the proof of the lemmas below since they follow directly the steps used in Agarwal et al. (2021) with $\nabla_\theta V^\pi(s_0)$ replaced with $g_t(\theta_t)$ and exploiting the vanishing property of $g_t(\theta_t)$ established in Part 1.

Lemma B.6. *[Lemma 47 of Agarwal et al. (2021)] Suppose $a_+ \in I_s^+$. For any $a \in I_s^0$, if there exists a $t > T_0$, such that $\pi_t(a_+|s) \geq \pi_t(a|s)$, then for all $\tau > t$, $\pi_\tau(a_+|s) \geq \pi_\tau(a|s)$.*

Therefore, for any action $a_+ \in I_s^+$, we can partition the set I_s^0 into $B_s^0(a_+)$ and $\bar{B}_s^0(a_+)$ such that $\forall a \in B_s^0(a_+)$, we have that $\pi_t(a_+|s) < \pi_t(a|s)$ for all $t \geq T_0$, and $\bar{B}_s^0(a_+)$ is the complement, i.e. $\forall a \in \bar{B}_s^0(a_+)$, there exists a $t' > T_0$ such that $\pi_t(a_+|s) \geq \pi_t(a|s)$ for all $t \geq t'$. In addition, for any action $a_+ \in I_s^+$, the set $B_s^0(a_+) \neq \emptyset$, as stated in the following lemma together with properties about $\pi_t(a|s)$ and $\theta_t(s,a)$ for $a \in B_s^0(a_+)$.

Lemma B.7. *[Lemma 48 of Agarwal et al. (2021)] Suppose $I_s^+ \neq \emptyset$. For all $a_+ \in I_s^+$, we have that $B_s^0(a_+) \neq \emptyset$ and that*

$$\sum_{a \in B_s^0(a_+)} \pi_t(a|s) \to 1, \text{as } t \to \infty$$

This implies that

$$\max_{a \in B_s^0(a_+)} \theta_t(s,a) \to \infty.$$

The following lemma also establishes useful properties about the sequence $\theta_t(s,a)$ depending on whether a is in I_s^+ or in I_s^-.

Lemma B.8. *[Lemma 50 of Agarwal et al. (2021)] For all actions $a_+ \in I_s^+$, the sequence $\theta_t(s,a)$ is bounded from below as $t \to \infty$. Moreover, for all actions $a_- \in I_s^-$, we have that $\theta_t(s,a) \to -\infty$ as $t \to \infty$.*

Based on Lemma B.8, we must have that for any $a_- \in I_s^-$, $\frac{\pi(a_-|s)}{\pi(a_+|s)} = \exp(\theta(s,a_-) - \theta(s,a_+)) \to 0$ since $\theta(s,a_+)$ is lower bounded and $\theta(s,a_-) \to -\infty$. There must therefore exist a $T_2 > T_0$ such that for all $t > T_2$

$$\frac{\pi_t(a_-|s)}{\pi_t(a_+|s)} < \frac{(1-\gamma)\Delta}{16|\mathcal{A}|},$$

or, equivalently,

$$-\sum_{a \in I_s^-} \frac{\pi_t(a_-|s)}{1-\gamma} > -\pi_t(a_+|s)\frac{\Delta}{16}. \tag{12}$$

For $\bar{a} \in \bar{B}_s^0$, we have $A^t(s,\bar{a}) \to 0$ by the definition of I_s^0. From Lemma B.6, there must exist a T_{a_+} such that when $t > T_{a_+}$, $\pi_t(a_+|s) \geq \pi_t(\bar{a}|s), \forall \bar{a} \in \bar{B}_s^0$. Thus, there exists a $T_3 > \max(T_2, T_{a_+})$ such that

$$|A^t(s,\bar{a})| < \frac{\Delta}{16|\mathcal{A}|} \leq \frac{\pi_t(a_+|s)}{\pi_t(\bar{a}|s)}\frac{\Delta}{16|\mathcal{A}|}, \ \forall \bar{a} \in \bar{B}_s^0, \ \forall t > T_3,$$

which implies

$$\sum_{\bar{a} \in \bar{B}_s^0(a_+)} \pi_t(\bar{a}|s)|A^t(s,\bar{a})| < \pi_t(a_+|s)\frac{\Delta}{16}, \ \forall t > T_3.$$

We thus obtain

$$-\pi_t(a_+|s)\frac{\Delta}{16} < \sum_{\bar{a} \in \bar{B}_s^0(a_+)} \pi_t(\bar{a}|s)A^t(s,\bar{a}) < \pi_t(a_+|s)\frac{\Delta}{16}, \ \forall t > T_3. \tag{13}$$

Putting the different bounds together, we get that for all $t > T_3$,

$$
\begin{aligned}
0 &= \sum_{a\in\mathcal{A}} \pi_t(a|s)A^t(s,a) \\
&= \sum_{a\in I_s^0} \pi_t(a|s)A^t(s,a) + \sum_{a_+\in I_s^+} \pi_t(a_+|s)A^t(s,a_+) + \sum_{a_-\in I_s^-} \pi_t(a_-|s)A^t(s,a_-) \\
&\overset{(a)}{\geq} \sum_{a\in B_s^0(a_+)} \pi_t(a|s)A^t(s,a) + \sum_{\bar{a}\in \bar{B}_s^0(a_+)} \pi_t(\bar{a}|s)A^t(s,\bar{a}) + \pi_t(a_+|s)A^t(s,a_+) \\
&\qquad\qquad + \sum_{a_-\in I_s^-} \pi_t(a_-|s)A^t(s,a_-) \\
&\overset{(b)}{\geq} \sum_{a\in B_s^0(a_+)} \pi_t(a|s)A^t(s,a) + \sum_{\bar{a}\in \bar{B}_s^0(a_+)} \pi_t(\bar{a}|s)A^t(s,\bar{a}) + \pi_t(a_+|s)\frac{\Delta}{4} - \sum_{a_-\in I_s^-} \frac{\pi_t(a_-|s)}{1-\gamma} \\
&\overset{(c)}{>} \sum_{a\in B_s^0} \pi_t(a|s)A^t(s,a) - \pi_t(a_+|s)\frac{\Delta}{16} + \pi^t(a_+|s)\frac{\Delta}{4} - \pi_t(a_+|s)\frac{\Delta}{16} \\
&> \sum_{a\in B_s^0(a_+)} \pi_t(a|s)A^t(s,a),
\end{aligned}
$$

where (a) uses $A^t(s,a_+) > 0, \forall a_+ \in I_s^+$ and only one a_+ is chosen; (b) uses $A^t(s,a_+) \geq \frac{\Delta}{4}$ for $t > T_1$ from Lemma B.5, and $A^t(s,a_-) \geq -\frac{1}{1-\gamma}$; (c) uses Eq. 12 and left inequality in Eq. 13. This implies that for all $t > T_3$

$$
\sum_{a\in B_s^0(a_+)} [g_t(\theta_t)](s,a) = \sum_{a\in B_s^0(a_+)} d(s)\pi_t(a|s)A^t(s,a) < 0. \tag{14}
$$

Since for $a \in B_s^0(a_+), \pi_t(a_+|s) < \pi_t(a|s)$ for all $t > T_0$. This implies by the softmax parameterization that $\theta_t(s,a_+) < \theta_t(s,a)$. Since $\theta_t(s,a_+)$ is lower bounded (Lemma B.8), this implies $\theta_t(s,a)$ is also lower bounded. This in conjunction with $\max_{a\in B_s^0(a_+)} \theta_t(s,a) \to \infty$ (Lemma B.7) implies that $\sum_{a\in B_s^0(a_+)} \theta_t(s,a) \to \infty$. This contradicts with Eq. 14 that the sum of updates for these same parameters is negative for all $t > T_3$. This contradiction completes the proof that $I_s^+ = \emptyset$.

Remark B.9. *Comparing with Agarwal et al. (2021), our theoretical results in Theorem 3.2 and 3.3 do not follow straightforwardly from the risk neutral results in Agarwal et al. (2021) given that our algorithm follows updates that cannot be interpreted as the gradient of the value function. We must therefore carefully adapt the proof of Agarwal et al. (2021) to account for our proxy gradient. The most important distinctions occur in the treatment of Lemma 41 and 44 in the original text. When adapting Lemma 41, we do not rely on performance difference lemma but instead on the monotonicity of the Bellman operator $\mathcal{T}^\pi$. In Lemma 44, we cannot rely on smoothness of V^{π_θ} with respect to θ and optimization theory to prove that our proxy gradient converges to zero. Instead, Lemma B.4 makes use of i) the β-smoothness and boundedness of the surrogate objective function from which our surrogate gradient is derived, and ii) the monotonicity of the Bellman operator $\mathcal{T}^\pi$. These modifications are non-trivial, and they extend Agarwal et al.'s guarantees from the risk-neutral objective to any coherent risk measure. Moreover, given that expected value is a coherent risk measure, our result implies that the Theorem 10 in Agarwal et al. (2021) also applies when policy gradient is applied off-policy (i.e. when their equation (10) is modified to use an arbitrary strictly positive $d(s)$ instead of the discounted state visitation distribution).*

C Proofs for Section 3.2

C.1 Proof of Proposition 3.4

The convergence analysis follows the framework presented in Sec. 4 of Bertsekas & Tsitsiklis (1996) and is similar to Shen et al. (2014). We first recall the standard operator convergence results in Bertsekas & Tsitsiklis (1996).

Consider the update rule

$$x_{t+1}(i) = (1 - \beta_t(i))x_t(i) + \beta_t(i)[(Hx_t)(i) + w_t(i)], \tag{15}$$

where $x_t \in \mathbb{R}^d$, $H : \mathbb{R}^d \to \mathbb{R}^d$ is an operator, w_t denotes some random noise and β_t is learning rate. Denote by $\mathcal{F}_t$ the history of the update up to time t,

$$\mathcal{F}_t := \{x_0, ..., x_t, w_0, ..., w_t, \beta_0, ..., \beta_t\}$$

Proposition C.1. *[Proposition 4.4 of Bertsekas & Tsitsiklis (1996)] Let x_t be the sequence generated by the iteration 15. Assume that*

1. The learning rates $\beta_t(i)$ are non-negative and satisfy

$$\sum_{t=0}^{\infty} \beta_t(i) = \infty, \ \sum_{t=0}^{\infty} \beta_t^2(i) < \infty, \ \forall i$$

2. The noise terms $w_t(i)$ satisfy 1) for all i and t, $\mathbb{E}[w_t(i)|\mathcal{F}_i] = 0$; 2) There exists a norm $||\cdot||$ on $\mathbb{R}^d$ and $a, b \in \mathbb{R}$ such that

$$\mathbb{E}[w_t^2(i)|\mathcal{F}_t] \leq a + b||Q_t||^2.$$

3. The operator H in 15 is a contraction under sup-norm.

Then x_t converges to x^*, a fixed point of H (i.e., $Hx^* = x^*$) with probability 1.

To prove Proposition 3.4, we first reformulate the updating rule in Eq. 10 as

$$Q_{t+1}(s, a) = Q_t(s, a) + 2\zeta_t(s, a) \cdot u(r + \gamma V_t(s') - Q_t(s, a)),$$

where $u(x) = (1 - \alpha)(x)_- + \alpha(x)_+$ with $\alpha \in (0, 1)$. It holds that $0 < \epsilon < \frac{u(x)-u(y)}{x-y} \leq L$ with $\epsilon := \min(\alpha, 1 - \alpha)$ and $L := \max(\alpha, 1 - \alpha)$ when $x \neq y$.

To align with Eq. 15, the update rule can be further expressed as

$$Q_{t+1}(s, a) = (1 - \frac{2\zeta_t(s, a)}{l})Q_t(s, a) + \frac{2\zeta_t(s, a)}{l}[l \cdot u(r + \gamma V_t(s') - Q_t(s, a)) + Q_t(s, a)],$$

where $l \in (0, \min(L^{-1}, 1)]$ (The range of l is chosen to obtain the contraction of H^π in the following).

With this formulation, define the operator H^π as

$$\begin{aligned}(H^\pi Q)(s, a) &= l \cdot \sum_{r,s'} P(r, s'|s, a)u(r + \gamma V(s') - Q(s, a)) + Q(s, a) \\ &\text{with } V(s') = \sum_{a'} \pi(a'|s')Q(s', a'),\end{aligned} \tag{16}$$

and noise w_t as

$$w_t(s, a) = l \cdot u(r + \gamma V_t(s') - Q_t(s, a)) - l \cdot \mathbb{E}_{r,s'}\big[u(r + \gamma V_t(s') - Q_t(s, a))\big]. \tag{17}$$

First, show that H^π in Eq. 16 is a contraction. We start with

$$\begin{aligned}|V_1(s) - V_2(s)| &= |\sum_a \pi(a|s)(Q_1(s, a) - Q_2(s, a))| \leq \sum_a \pi(a|s)|Q_1(s, a) - Q_2(s, a)| \\ &\leq \sum_a \pi(a|s)\|Q_1 - Q_2\|_\infty = \|Q_1 - Q_2\|_\infty.\end{aligned}$$

Next, we can show

$$
\begin{aligned}
(H^\pi Q_1)(s,a) - (H^\pi Q_2)(s,a) =& l \cdot \sum_{r,s'} P(r,s'|s,a)\big(u(r+\gamma V_1(s') - Q_1(s,a)) - u(r+\gamma V_2(s') - Q_2(s,a))\big) \\
&+ Q_1(s,a) - Q_2(s,a) \\
=& l \cdot \sum_{r,s'} P(r,s'|s,a)\xi_{(s,a,r,s',Q_1,Q_2)}\big(\gamma V_1(s') - \gamma V_2(s') - Q_1(s,a) + Q_2(s,a)\big) \\
&+ Q_1(s,a) - Q_2(s,a) \\
\overset{(a)}{=}& l\gamma \sum_{r,s'} P(r,s'|s,a)\xi_{(s,a,r,s',Q_1,Q_2)}(V_1(s') - V_2(s')) \\
&+ \big(1 - l\sum_{r,s'} P(r,s'|s,a)\xi_{(s,a,r,s',Q_1,Q_2)}\big)\big(Q_1(s,a) - Q_2(s,a)\big) \\
\overset{(b)}{\leq}& l\gamma \sum_{r,s'} P(r,s'|s,a)\xi_{(s,a,r,s',Q_1,Q_2)}\|Q_1 - Q_2\|_\infty \\
&+ \big(1 - l\sum_{r,s'} P(r,s'|s,a)\xi_{(s,a,r,s',Q_1,Q_2)}\big)\|Q_1 - Q_2\|_\infty \\
=& \big(1 - l(1-\gamma)\sum_{r,s'} P(r,s'|s,a)\xi_{(s,a,r,s',Q_1,Q_2)}\big)\|Q_1 - Q_2\|_\infty \\
\leq& (1 - l(1-\gamma)\epsilon)\|Q_1 - Q_2\|_\infty,
\end{aligned}
$$

where (a) is due to $0 < \epsilon < \frac{u(x)-u(y)}{x-y} \leq L$, so there exists $\xi_{(x,y)} \in (\epsilon, L]$ such that $u(x) - u(y) = \xi_{(x,y)}(x-y)$; (b) is due to $V_1(s') - V_2(s') \leq \|Q_1 - Q_2\|_\infty$ and $\big(1 - l\sum_{r,s'} P(r,s'|s,a)\xi_{(s,a,r,s',Q_1,Q_2)}\big) \geq 0$ when $l \in (0, \min(L^{-1}, 1)]$. Therefore, H^π is a contraction.

Q^π is the fixed point of H^π, since when $Q = Q^\pi$, the gradient of the loss function $(1-\alpha)\mathbb{E}\big[(\tilde{r} + \gamma V(\tilde{s}') - y)_-^2\big] + \alpha\mathbb{E}\big[(\tilde{r} + V(\tilde{s}') - y)_+^2\big]$ is zero, which is $\mathbb{E}_{r,s'}[u(r+\gamma V(s') - Q(s,a))] = 0$. As a result, Eq. 16 remains $H^\pi Q^\pi = Q^\pi$.

Second, check the conditions of the noise term $w_t(s,a)$ in Eq. 17. $\mathbb{E}[w_t(s,a)|\mathcal{F}_t] = 0$ by its definition, thus condition 1) holds. Denote $\delta_t(s,a,r,s') = r + \gamma V_t(s') - Q_t(s,a)$,

$$
\mathbb{E}\big[w_t^2(s,a)|\mathcal{F}_t\big] = l^2\mathbb{E}_{r,s'}\big[u^2(\delta_t(s,a,r,s'))\big] - l^2\big(\mathbb{E}_{r,s'}\big[u(\delta_t(s,a,r,s'))\big]\big)^2 \leq l^2\mathbb{E}_{r,s'}\big[u^2(\delta_t(s,a,r,s'))\big]
$$

Let $\bar{R}$ be the bound of $|\tilde{r}|$. Then $|\delta_t(s,a,r,s')| \leq \bar{R} + 2\|Q_t\|_\infty$, which implies $|u(\delta_t(s,a,r,s')) - u(0)| \leq L|\delta_t(s,a,r,s') - 0| \leq L(\bar{R} + 2\|Q_t\|_\infty)$. Therefore

$$
\big(u(\delta_t(s,a,r,s'))\big)^2 = \big(\big|u(\delta_t(s,a,r,s')) - u(0)\big|\big)^2 \leq \big(L\bar{R} + 2L\|Q_t\|_\infty\big)^2 \leq 2\big(L\bar{R}\big)^2 + 8L^2\|Q_t\|_\infty^2,
$$

where the last inequality is due to Cauchy-Schwarz inequality $(x+y)^2 \leq 2x^2 + 2y^2$. As a result

$$
\mathbb{E}\big[w_t^2(s,a)|\mathcal{F}_t\big] \leq l^2 \cdot \mathbb{E}_{r,s'}\big[u^2(\delta_t(s,a,r,s'))\big] \leq 2l^2\big(L\bar{R}\big)^2 + 8l^2L^2\|Q_t\|_\infty^2.
$$

Hence, condition 2) holds.

The contraction of H^π and the conditions being satisfied by $w_t(s,a)$ indicate $Q_t(s,a) \to Q^\pi(s,a)$, which completes the proof for Proposition 3.4.

C.2 Proof of Proposition 3.5

The convergence analysis follows the two-time scale framework presented in Chapter 8 of Borkar (2023). We first recall this convergence results.

Consider the iterations

$$
\begin{aligned}
x_{n+1} &= x_n + a_n[h(x_n, y_n) + M_{n+1}^{(1)}], \\
y_{n+1} &= y_n + b_n[g(x_n, y_n) + M_{n+1}^{(2)}],
\end{aligned}
\tag{18}
$$

where $h : \mathbb{R}^{d_x+d_y} \to \mathbb{R}^{d_x}, g : \mathbb{R}^{d_x+d_y} \to \mathcal{R}^{d_y}$ (d_x and d_y represent the dimension of x and y) are Lipschitz, and $\{M_n^{(1)}\}, \{M_n^{(2)}\}$ are martingale difference sequences with respect to the increasing σ-fields

$$\mathcal{F}_n := \sigma(x_m, y_m, M_m^{(1)}, M_m^{(2)}, m \le n), n \ge 0,$$

satisfying

$$\mathbb{E}[\|M_{n+1}^{(i)}\|^2 | \mathcal{F}_n] \le K(1 + \|x_n\|^2 + \|y_n\|^2), i = 1, 2.$$

Stepsizes $\{a_n\}, \{b_n\}$ are positive scalars satisfying

$$\sum_t a_n = \sum_t b_n = \infty, \sum_t (a_n^2 + b_n^2) < \infty, \frac{b_n}{a_n} \to 0.$$

Proposition C.2. *[Theorem 8.1 in Chapter 8 of Borkar (2023)] Let x_n, y_n be the sequences generated by iterations 18. Assume that*

1. $\dot{x}(t) = h(x(t), y)$ has a globally asymptotically stable equilibrium $\lambda(y)$, where $\lambda : \mathbb{R}^{d_y} \to \mathbb{R}^{d_x}$ is a Lipschitz map.

2. $\dot{y}(t) = g(\lambda(y(t)), y(t))$ has a globally asymptotically stable equilibrium y^*.

3. $\sup_n(\|x_n\| + \|y_n\|) < \infty$ almost surely.

Then $(x_n, y_n) \to (\lambda(y^*), y^*)$ almost surely.

We reformulate the update rule for q and Q in Eq. 11 to align with Eq. 18. To distinguish with the t in ODE, we use n as the subscript for q and Q instead.

$$\begin{aligned}
q_{n+1}(s,a) &= q_n(s,a) + \zeta_n^q(s,a)\big(h_{s,a}(q_n, Q_n) + M_{n+1}^q(s,a)\big), \text{ with} \\
h_{s,a}(q_n, Q_n) &= -\Big(\mathbb{P}\big(\tilde{r}(s,a) + \gamma V_n(\tilde{s}') < q_n(s,a)\big) - \alpha\Big), \\
M_{n+1}^q(s,a) &= \mathbb{P}\big(\tilde{r}(s,a) + \gamma V_n(\tilde{s}') < q_n(s,a)\big) - \mathbb{I}\{r + \gamma V_n(s') < q_n(s,a)\}.
\end{aligned}$$

$$\begin{aligned}
Q_{n+1}(s,a) &= Q_n(s,a) + \zeta_n^Q(s,a)(g_{s,a}(q_n, Q_n) + M_{n+1}^Q(s,a)), \text{ with} \\
g_{s,a}(q_n, Q_n) &= -\Big(Q_n(s,a) - q_n(s,a) + \frac{1}{\alpha}\mathbb{E}_{r,s'}\big[(q_n(s,a) - r - \gamma V_n(s'))_+\big]\Big), \\
M_{n+1}^Q(s,a) &= \frac{1}{\alpha}\mathbb{E}_{r,s'}\big[(q_n(s,a) - r - \gamma V_n(s'))_+\big] - \frac{1}{\alpha}(q_n(s,a) - r - \gamma V_n(s'))_+,
\end{aligned}$$

where $V_n(s') = \sum_{a'} \pi(a'|s') Q_n(s,a)$.

First, check the Lipschitz of h and g.

For h, when Q is fixed, since we assume reward has bounded density, $f_{\tilde{r}(s,a)+\gamma V(\tilde{s}')}(z) < M$, then

$$\begin{aligned}
|h_{s,a}(q_1, Q) - h(q_2, Q)| &= |F_{\tilde{r}(s,a)+\gamma V(\tilde{s}')}(q_2(s,a)) - F_{\tilde{r}(s,a)+\gamma V(\tilde{s}')}(q_1(s,a))| \\
&= \Big| \int_{q_1(s,a)}^{q_2(s,a)} f_{\tilde{r}(s,a)+\gamma V(\tilde{s}')}(z) dz \Big| \\
&\le M|q_1 - q_2|,
\end{aligned}$$

where F is the CDF and f is the density function.

When q is fixed,

$$\begin{aligned}
|h_{s,a}(q, Q_1) - h_{s,a}(q, Q_2)| &= |\mathbb{P}(\tilde{r}(s,a) + \gamma V_1(\tilde{s}') < q(s,a)) - \mathbb{P}(\tilde{r}(s,a) + \gamma V_2(\tilde{s}') < q(s,a))| \\
&= |F_{\tilde{r}(s,a)+\gamma V_1(\tilde{s}')}(q(s,a)) - F_{\tilde{r}(s,a)+\gamma V_2(\tilde{s}')}(q(s,a))|
\end{aligned}$$

Denote $\tilde{z}_1 := \tilde{r}(s,a) + \gamma V_1(\tilde{s}')$, and $\tilde{z}_2 := \tilde{r}(s,a) + \gamma V(\tilde{s}')$, we have $|\tilde{z}_1 - \tilde{z}_2| = |\gamma V_1(\tilde{s}') - \gamma V_2(\tilde{s}')| \le \|Q_1 - Q_2\|_\infty := \epsilon$ almost surely. Thus $\tilde{z}_2 < q - \epsilon$ ensures $\tilde{z}_1 < q$, which implies $F_{\tilde{z}_2}(q - \epsilon) \le F_{\tilde{z}_1}(q)$. By the same logic, we have $F_{\tilde{z}_1}(q) \le F_{\tilde{z}_2}(q + \epsilon)$. This leads to

$$|F_{\tilde{z}_1}(q(s,a)) - F_{\tilde{z}_2}(q(s,a))| \le |F_{\tilde{z}_2}(q(s,a) + \epsilon) - F_{\tilde{z}_2}(q(s,a))| \le M\|Q_1 - Q_2\|_\infty.$$

As a result,

$$\begin{aligned}|h_{s,a}(q_1,Q_1)-h_{s,a}(q_2,Q_2)| &\le |h_{s,a}(q_1,Q_1)-h_{s,a}(q_2,Q_1)|+|h_{s,a}(q_2,Q_1)-h_{s,a}(q_2,Q_2)|\\ &\le M|q_1-q_2|+M\|Q_1-Q_2\|_\infty\end{aligned}$$

For g, we need to show its third term, i.e., $k_{s,a}(q,Q):=\mathbb{E}_{r,s'}[(q(s,a)-r-\gamma V(s'))_+]$ is Lipschitz.

$$\begin{aligned}|k_{s,a}(q_1,Q_1)-k_{s,a}(q_2,Q_2)| =&|\mathbb{E}[(q_1(s,a)-r-\gamma V_1(s'))_+]-\mathbb{E}[(q_2(s,a)-r-\gamma V_2(s'))]|\\ \le&\mathbb{E}[|q_1-q_2-\gamma(V_1(s')-V_2(s'))|]\\ \le&|q_1-q_2|+\gamma\mathbb{E}[|V_1(s')-V_2(s')|]\\ \le&|q_1-q_2|+\|Q_1-Q_2\|_\infty\end{aligned}$$

Second, check the conditions of the difference terms.

$|M^q_{n+1}(s,a)|\le 1$, thus $\mathbb{E}[M^q_{n+1}(s,a)^2|\mathcal{F}_n]\le 1$. Denote $\delta_n(s,a,r,s')=q_n(s,a)-r-\gamma V_n(s')$,

$\mathbb{E}[M^Q_{n+1}(s,a)^2|\mathcal{F}_n]=\frac{1}{\alpha^2}\mathbb{E}_{r,s'}[(\delta_n(s,a,r,s'))^2_+]-\frac{1}{\alpha^2}\big(\mathbb{E}_{r,s'}\big[(\delta_n(s,a,r,s'))_+\big]\big)^2\le\frac{1}{\alpha^2}\mathbb{E}_{r,s'}\big[\big(\delta_n(s,a,r,s')\big)^2_+\big]$.

Let $\bar R$ be the bound of $|\tilde r|$. Since $\big(\delta_n(s,a,r,s')\big)_+\le|\delta_n(s,a,r,s')|=|q_n(s,a)-r-\gamma V(s')|\le\|q_n\|_\infty+\bar R+\|Q_n\|_\infty$,

$$\mathbb{E}[M^Q_{n+1}(s,a)^2|\mathcal{F}_n]\le\frac{1}{\alpha^2}\mathbb{E}_{r,s'}\big[\big(\delta_n(s,a,r,s')\big)^2_+\big]\le\frac{3}{\alpha^2}(\bar R^2+\|q_n\|^2_\infty+\|Q_n\|^2_\infty),$$

where the last inequality follows Cauchy-Schwarz inequality $(x+y+z)^2<=3(x^2+y^2+z^2)$.

Therefore, there exists K, such that $\mathbb{E}[M_{n+1}(s,a)^2|\mathcal{F}_n]\le K(1+\|q_n\|^2_\infty+\|Q_n\|^2_\infty)$ for both M^q_{n+1} and M^Q_{n+1}.

Third, check the other three conditions in Proposition C.2. Recall that we assume the random variable is continuous and the quantile is unique in Sec. 2.1.

1) When Q is fixed (therefore V is also fixed), for $\dot q_{s,a}(t)=-\Big(\mathbb{P}\big(\tilde r(s,a)+\gamma V(\tilde s')\le q_{s,a}(t)\big)-\alpha\Big)$, setting $\dot q_{s,a}(t)=0$ results in the equilibrium $q^*(s,a)=\mathrm{VaR}_\alpha(\tilde r(s,a)+\gamma V(\tilde s'))$. Denote $\tilde z:=\tilde r(s,a)+\gamma V(\tilde s')$. It is clear that when $q_{s,a}(t)<q^*(s,a)$, $\dot q_{s,a}(t)>0$ and when $q_{s,a}(t)>q^*(s,a)$, $\dot q_{s,a}(t)<0$. Consider the case when $q_{s,a}(0)<q^*(s,a)$. Since $\dot q_{s,a}(t)>0$, and $q_{s,a}(t)$ is upper bounded by $q^*(s,a)$, we have $\lim_{t\to\infty}q_{s,a}(t)=\bar q(s,a)$ with $\bar q(s,a)\le q^*(s,a)$. Suppose $\bar q(s,a)<q^*(s,a)$, then $\exists\epsilon>0$, $\dot q_{s,a}(t)=\alpha-F_{\tilde z}\big(q_{s,a}(t)\big)>\alpha-F_{\tilde z}\big(\bar q(s,a)\big)>\epsilon$, where $F_{\tilde z}$ is a CDF with unique quantiles. Therefore, for all $t\ge0$, $q_{s,a}(t)\ge q_{s,a}(0)+\epsilon t$. Since $\lim_{t\to\infty}\big(q_{s,a}(0)+\epsilon t\big)\to\infty$, this contradicts the fact that $\lim_{t\to\infty}q_{s,a}(t)=\bar q(s,a)\le q^*(s,a)$. Hence, it can only be the case that $\lim_{t\to\infty}q_{s,a}(t)=\bar q(s,a)=q^*(s,a)$. The same argument holds when $q_{s,a}(0)>q^*(s,a)$. This indicates the global attractivity of $q^*(s,a)$.

For the Lyapunov stability, define the Lyapunov function

$$\mathscr{L}(q)=\int_{q^*}^{q}(F_{\tilde z}(z)-\alpha)dz.$$

We have $\mathscr{L}(q)\ge0$ and $\mathscr{L}(q)=0$ only if $q=q^*$, and $\dot{\mathscr{L}}(q_{s,a}(t))=-(F_{\tilde z}(q_{s,a}(t))-\alpha)^2\le0$ for all t, which verifies the stability.

The global attractivity of q^* and Lyapunov stability together imply that q^* is globally asymptotically stable.

For the λ function, in our case,

$$\lambda_{s,a}(Q)=\mathrm{VaR}_\alpha(\tilde r(s,a)+\gamma\mathbb{E}_{a'\sim\pi(\cdot|\tilde s')}[Q(\tilde s',a')]).$$

we have for all (s,a)

$$\begin{aligned}&|\lambda_{s,a}(Q_1)-\lambda_{s,a}(Q_2)|\\ =&|\mathrm{VaR}_\alpha(\tilde r(s,a)+\gamma\mathbb{E}_{a'\sim\pi(\cdot|\tilde s')}[Q_1(\tilde s',a')])-\mathrm{VaR}_\alpha(\tilde r(s,a)+\gamma\mathbb{E}_{a'\sim\pi(\cdot|\tilde s')}[Q_2(\tilde s',a')])|\\ \le&\max_{s'}|\gamma\mathbb{E}_{a'\sim\pi(\cdot|s')}[Q_1(s',a')]-\gamma\mathbb{E}_{a'\sim\pi(\cdot|s')}[Q_2(s',a')]|\\ \le&\gamma\|Q_1-Q_2\|_\infty,\end{aligned}$$

which means $\lambda(\cdot)$ is Lipschitz.

2) When $q_{s,a}(t) = \mathrm{VaR}_\alpha(\tilde{r}(s,a) + \gamma V_{\tilde{s}'}(t))$,

$$\dot{Q}_{s,a}(t) = q_{s,a}(t) - \frac{1}{\alpha}\mathbb{E}_{r,s'}[(q_{s,a}(t) - r - \gamma V_{s'}(t))_+] - Q_{s,a}(t) = (\mathcal{T}^\pi Q(t))(s,a) - Q_{s,a}(t). \quad (19)$$

Setting $\dot{Q}(t) = 0$, the equilibrium satisfies $\mathcal{T}^\pi Q = Q$, which is Q^π. Showing Q^π is globally asymptotically stable relies on the following theorem in Borkar (2023)

Theorem C.3 (Theorem 12.1 in Chapter 12 of Borkar (2023))**.** *Consider iterations of the type*

$$x_{n+1} = x_n + a_n[F(x_n) - x_n + M_{n+1}].$$

For $x = [x_1, ..., x_d] \in \mathbb{R}^d$ and $w = [w_1, ..., w_d]$, define

$$\|x\|_{w,p} \stackrel{def}{=} \Big(\sum_{i=1}^d w_i |x_i|^p\Big)^{1/p}, 1 \le p < \infty,$$

and

$$\|x\|_{w,\infty} \stackrel{def}{=} \max_i (w_i |x_i|).$$

Assume that

$$\|F(x) - F(y)\|_{w,p} \le \alpha \|x - y\|_{w,p}, \forall x, y \in \mathbb{R}^d, \ \text{(contraction)}$$

for some w, $1 \le q < \infty$ and $\alpha \in (0,1)$.

With the contraction of F, if the limiting ODE is $\dot{x}(t) = F(x(t)) - x(t)$, with unique x^* such that $x^* = F(x^*)$, then x^* is the globally asymptotically stable equilibrium.

In our case, $\mathcal{T}^\pi$ is a contraction under the sup-norm and Q^π is the unique solution of $Q = \mathcal{T}^\pi(Q)$. Therefore, according to Theorem C.3, Q^π is the globally asymptotically stable equilibrium of equation 19.

3) $\sup_t(\|q_t\|_\infty + \|Q_t\|_\infty) < \infty$ holds almost surely by our assumption.

All the above analysis completes the proof for Proposition 3.5.

D Discussion on the Expectile Bellman Operator in Jiang et al. (2024)

For a policy $\pi(\cdot|s)$, the Expectile Bellman operator proposed by Jiang et al. (2024) is

$$\mathcal{T}^\pi V(s) = V(s) + 2\zeta \mathbb{E}_{a\sim\pi(\cdot|s)}\mathbb{E}_{r,s'}\big[(1-\alpha)\big(\delta(s,a,r,s')\big)_- + \alpha\big(\delta(s,a,r,s')\big)_+\big],$$

where $\delta(s,a,r,s') = r(s,a) + \gamma V(s') - V(s)$. The contraction of this Bellman operator is established in Jiang et al. (2024), but its fixed point is not explicitly characterized.

For a specific action $a \sim \pi(\cdot|s)$, the policy is updated using the advantage $A^\pi(s,a) := \mathbb{E}_{r,s'}\big[(1-\alpha)\big(\delta(s,a,r,s')\big)_- + \alpha\big(\delta(s,a,r,s')\big)_+\big]$ in Jiang et al. (2024). This naturally motivates considering the optimality version of this Bellman operator, i.e.,

$$\mathcal{T}^* V(s) = V(s) + \max_a 2\zeta \mathbb{E}_{r,s'}\big[(1-\alpha)\big(\delta(s,a,r,s')\big)_- + \alpha\big(\delta(s,a,r,s')\big)_+\big]. \quad (20)$$

We show that the fixed point of this optimality Bellman operator is V^*, which suggests that the method in Jiang et al. (2024) optimizes dynamic expectile.

Lemma D.1. *V^* is the unique solution to the Bellman operator in Eq. 20.*

Proof. Following Eq. 5, the Bellman optimality equation for V^* is

$$V^*(s) = \max_a Q^*(s,a) = \max_a \rho(\tilde{r}(s,a) + \gamma V^*(\tilde{s}')). \quad (21)$$

By subtracting $V(s)$ from both side of Eq. 20 and dividing by 2ζ, we obtain

$$\max_a \mathbb{E}_{r,s'}\big[(1-\alpha)\big(\delta(s,a,r,s')\big)_- + \alpha\big(\delta(s,a,r,s')\big)_+\big] = 0 \quad (22)$$

Eq. 21 implies that for all a, we have that:

$$\begin{aligned} V^*(s) \geq & \rho(\tilde{r}(s,a) + \gamma\, V^*(\tilde{s}')) \\ & \stackrel{\text{def}}{=} \arg\min_y \mathbb{E}[L(\tilde{r}(s,a) + \gamma V^*(\tilde{s}'), y)],\ \forall a \\ & = \arg\min_y (1-\alpha)\mathbb{E}\big[\big(\tilde{r}(s,a) + \gamma V^*(\tilde{s}) - y\big)_-^2\big] + \alpha\mathbb{E}\big[\big(\tilde{r}(s,a) + \gamma V^*(\tilde{s}) - y\big)_+^2\big],\ \forall a \end{aligned}$$

which implies that for all a

$$\begin{aligned} 0 \leq & \nabla_y \mathbb{E}[L(\tilde{r}(s,a) + \gamma\, V^*(\tilde{s}'), y)]|_{y=V^*(s)} \\ & = -2\mathbb{E}[(1-\alpha)(\tilde{r}(s,a) + \gamma V^*(\tilde{s}') - V^*(s))_- + \alpha(\tilde{r}(s,a) + \gamma V^*(\tilde{s}') - V^*(s))_+], \quad \forall a. \end{aligned}$$

Therefore

$$\max_a \mathbb{E}[(1-\alpha)(\tilde{r}(s,a) + \gamma V^*(\tilde{s}') - V^*(s))_- + \alpha(\tilde{r}(s,a) + \gamma V^*(\tilde{s}') - V^*(s))_+] \leq 0. \tag{23}$$

Moreover, Eq. 5 implies that there exists an optimal action $\pi^*(s)$ such that

$$\begin{aligned} V^*(s) = & \rho(r(s,\pi^*(s)) + \gamma V^*(\tilde{s}')) \\ & = \arg\min_y \mathbb{E}\big[L\big(\tilde{r}(s,\pi^*(s)) + \gamma V^*(\tilde{s}'), y\big)\big]. \end{aligned}$$

Thus we can confirm that the maximum in Eq. 23 is 0 and is achieved when $a = \pi^*(s)$, i.e.,

$$\mathbb{E}[(1-\alpha)(\tilde{r}(s,\pi^*(s)) + \gamma V^*(\tilde{s}') - V^*(s))_- + \alpha(\tilde{r}(s,\pi^*(s)) + \gamma V^*(\tilde{s}') - V^*(s))_+] = 0.$$

As a result,

$$\max_a \mathbb{E}[(1-\alpha)(\tilde{r}(s,a) + \gamma V^*(\tilde{s}') - V^*(s))_- + \alpha(\tilde{r}(s,a) + \gamma V^*(\tilde{s}') - V^*(s))_+] = 0$$

□

E Experiment Details

E.1 Policy and Value Function Parameterization

For tabular domains (i.e., Maze, CliffWalk), states are represented by the coordinates, i.e. $s := (x, y)$. Policy parameter is represented by a matrix Θ with size $[n_row, n_column, 4]$. Action probability is calculated by $\pi(a|s) = \frac{\exp(\Theta[x,y,a])}{\sum_b \exp(\Theta[x,y,b])}$. Similarly, value function is also represented by a matrix Φ with size $[n_row, n_column, 4]$ (if learning state-action value) or $[n_row, n_column]$ (if learning state value).

For LunarLander, both policy and value functions are implemented by deep neural networks using `Pytorch` with two hidden layers and `nn.ReLU()` activation. The hidden layer dimension is 128. Softmax is applied to the output of policy network then feed to `torch.distributions.Categorical()`. The output dimension of value network is $n_actions$ (if learning state-action value) or 1 (if learning state value).

For Inverted Pendulum, both policy and value functions are implemented by deep neural networks using `Pytorch` with two hidden layers and `nn.ReLU()` activation. The hidden layer dimension is 128 (256 in HalfCheetah). `nn.Tanh()` is applied to the output of policy network then multiplied by the maximal magnitude of action to produce the policy mean. Std is a separate learning parameter using `nn.Parameter()`. The mean and std are feed to `torch.distributions.Normal()`. The input of value function is state plus action (if learning state-action value) or state only (if learning state value).

Specifically for Coache et al. (2023), as discussed in Sec. 3.4, the original method works on the right tail CVaR. To align with our setting, we replace its value updates by the two-time scale rules as in Eq. 11. Note that this method learns state-only value function.

Table 1: Learning parameters in Maze and CliffWalk.

Method	π lr	Q (or V) lr	q lr
Exp-AC	5e-3	5e-3	-
CVaR-AC	5e-4	1e-3	1e-2
Exp-PPO	5e-3	1e-2	-
QL	-	5e-3	-

Table 2: Learning parameters in LunarLander.

Method	π lr	Q (or V) lr	q lr
Exp-AC	7e-5	3.5e-4	-
CVaR-AC	1e-4	2e-4	1e-3
Exp-PPO	1e-4	5e-4	-
EPG	5e-5	2.5e-4	-

E.2 Learning Parameters

Maze and **CliffWalk**. The maze domain is modified from the Github repo [https://github.com/ido90/CeSoR] under MIT license. The cliffwalk domain is implemented by authors.

Discount factor $\gamma = 0.999$. For Exp-AC, CVaR-AC, and Exp-PPO, policy learning rate is searched from {5e-4, 1e-3, 5e-3}, value learning rate is searched from {1,2,5} times the policy learning rate. Batch size is 128 for Exp-AC, CVaR-AC, and QL. Exp-PPO collects 500 steps before updating policy. The learning rates selected are summarized in Table 1.

LunarLander. We use the discrete action version of LunarLander, which is modified from the Github repo [https://github.com/openai/gym] under MIT License. The state dimension is 8, the action space is 4. A detailed description is available at this webpage[1]. The maximum episode length is 500.

Discount factor $\gamma = 0.99$. Optimizer is Adam. Policy learning rate is searched from {3e-5, 5e-5, 7e-5, 1e-4, 3e-4, 5e-4}. Value learning rate is searched from {1,2,5} times the policy learning rate. For Exp-AC, CVaR-AC, and EPG, the entropy coefficient starts from 0.1 and linearly decays to 0.01 in the beginning 15% of steps. The entropy coefficient of Exp-PPO is 0.01. Batch size is 128 for Exp-AC, CVaR-AC, and EPG. Exp-PPO collected 1000 steps before updating policy. The learning rates selected are summarized in Table 2.

Inverted Pendulum. This domain is modified from the Github repo [https://github.com/openai/gym] under MIT License. The state dimension is 4. The one dimension action is continuous in range $[-3, 3]$. A detailed description is available at this webpage[2]. The maximum episode length is 300.

Discount factor $\gamma = 0.99$. Optimizer is Adam. For each update, Exp-AC, CVaR-AC, and EPG sample 10 actions to compute policy gradient and the backup value. Policy learning rate is searched from {3e-5, 5e-5, 7e-5, 1e-4, 3e-4, 5e-4}. Value learning rate is searched from {1,2,5} times the policy learning rate. For Exp-AC, CVaR-AC, and EPG, the entropy coefficient starts from 0.1 and linearly decays to 0.01 in the beginning 15% of steps. The entropy coefficient of Exp-PPO is 0.01. Batch size is 128 for Exp-AC, CVaR-AC, and EPG. Exp-PPO collects 1000 steps before updating policy. The learning rates selected are summarized in Table 3.

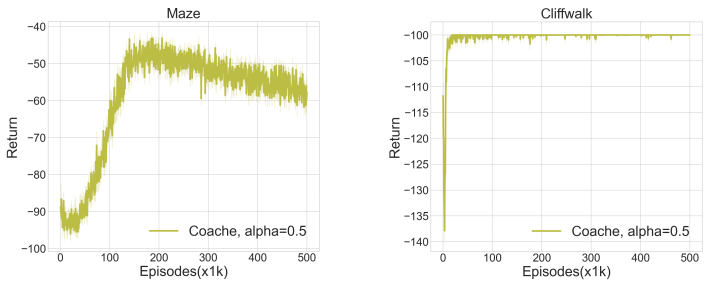
Coache et al. (2023). We fail to get a good performing policy within the same parameter search range. We report the best expected return learning curves of Coache et al. (2023) we obtained in Fig. 3. α is selected from $\{0.5, 0.7, 0.9\}$. The learning rate search range remains the same as other methods. The learning rates selected are summarized in Table 4.

[1] https://gymnasium.farama.org/environments/box2d/lunar_lander/

[2] https://gymnasium.farama.org/environments/mujoco/inverted_pendulum/

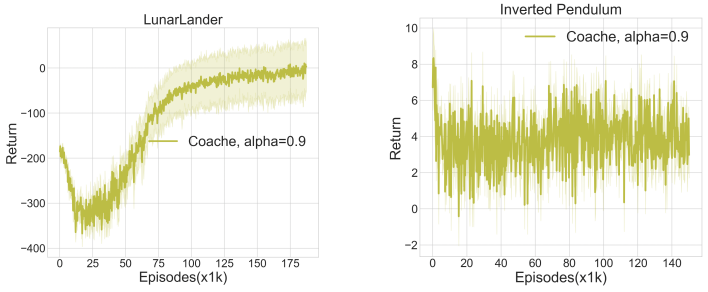
Table 3: Learning parameters in Inverted Pendulum.

Method	π lr	Q (or V) lr	q lr
Exp-AC	5e-5	2.5e-4	-
CVaR-AC	5e-5	2.5e-4	1e-3
Exp-PPO	5e-4	2.5e-3	-
EPG	7e-5	3.5e-4	-



(a) Expected return in maze

(b) Expected return in cliffwalk



(c) Expected return in LunarLander

(d) Expected return in Pendulum

Figure 3: Expected return of the meathod in Coache et al. (2023) in four evaluation domains. Curves are averaged over 10 seeds with shaded regions indicating standard errors.

Table 4: Learning parameters of Coache et al. (2023) in four domains.

Env	π lr	V lr	quantile lr
Maze	1e-4	2e-4	1e-3
Cliffwalk	1e-4	2e-4	1e-3
LunarLander	3e-5	6e-5	3e-4
Pendulum	5e-5	1e-4	5e-4

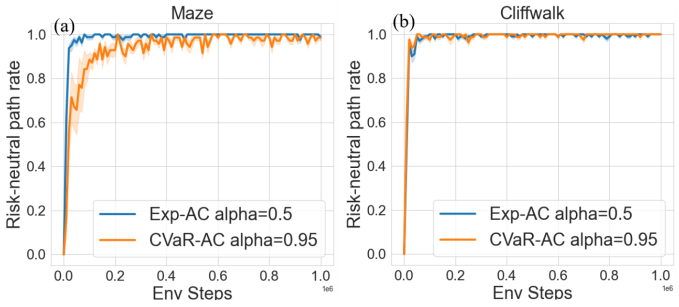
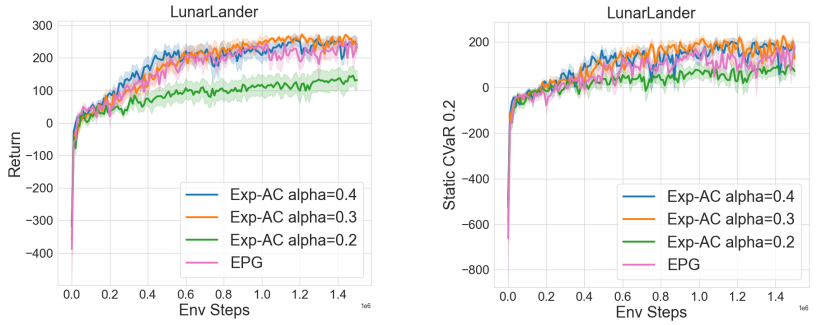
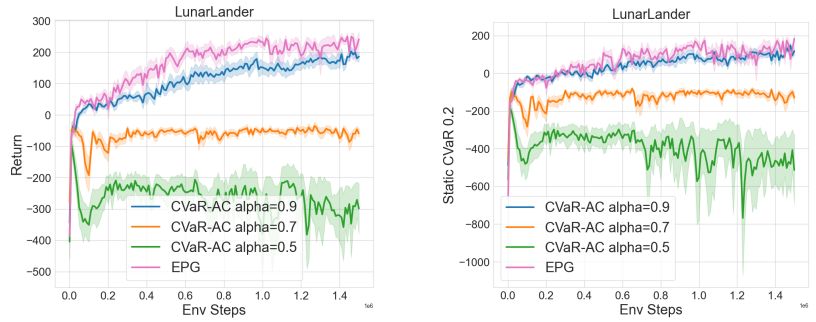


Figure 4: Risk-neutral path rate in Maze and Cliffwalk. Curves are averaged over 5 seeds with shaded regions indicating standard errors.



(a) Expected return in Lunarlander

(b) CVaR 0.2 of return in Lunarlander



(c) Expected return in Lunarlander

(d) CVaR 0.2 of return in Lunarlander

Figure 5: Expected return and CVaR 0.2 of return in LunarLander under different risk levels. Curves are averaged over 10 seeds with shaded regions indicating standard errors.

E.3 Additional Results

E.3.1 Curves under Different Risk Levels of Exp-AC and CVaR-AC

We report the learning curves of different risk levels of Exp-AC and CVaR-AC.

Maze and **Cliffwalk**. We report results on learning risk-neutral policies of Exp-AC and CVaR-AC. For Exp-AC, $\alpha = 0.5$ and for CVaR-AC, $\alpha = 0.95$ (we do not choose $\alpha = 1$ for CVaR-AC to avoid learning extreme quantiles). Agents are evaluated by sampling 10 trajectories during training and the risk-neutral path rate of sampled trajectories are shown in Fig 4. In Maze, for Exp-AC, π lr=5e-3, Q lr=5e-3; for CVaR-AC, π lr=5e-4, Q lr=5e-3, q lr=1e-2. In Cliffwalk, for Exp-AC, π lr=5e-3, Q lr=5e-3; for CVaR-AC, π lr=5e-3, Q lr=5e-3, q lr=1e-2.

LunarLander. The expected return and static 0.2-CVaR of the return under different risk levels of Exp-AC and CVaR-AC are shown in Fig. 5. For Exp-AC, when $\alpha = 0.4$, π lr=7e-5, Q lr=3.5e-4; when $\alpha = 0.2$, π lr=7e-5, Q lr=3.5e-4. For CVaR-AC, when $\alpha = 0.7$, π lr=5e-5, Q lr=1e-4, q lr=5e-4; when $\alpha = 0.5$, π lr=5e-5, Q lr=1e-4, q lr=2.5e-4. We also observe that Exp-AC achieves a high left landing rate as shown in Fig. 6 (corresponds to learning parameters in Table 2).

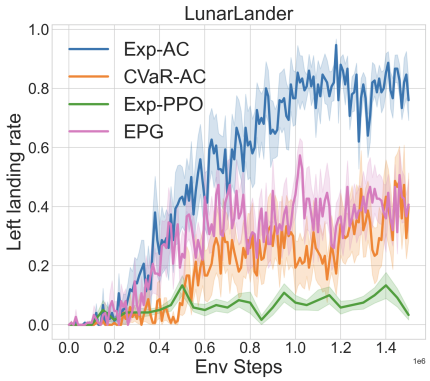
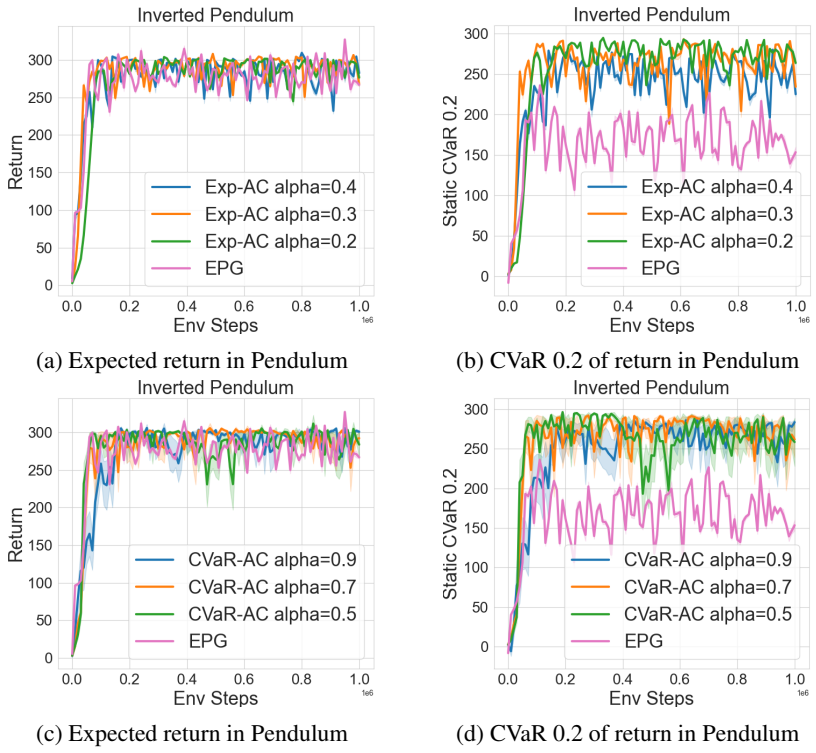


Figure 6: Left landing rates in LunarLander. Curves are averaged over 10 seeds with shaded regions indicating standard errors.



(a) Expected return in Pendulum

(b) CVaR 0.2 of return in Pendulum

(c) Expected return in Pendulum

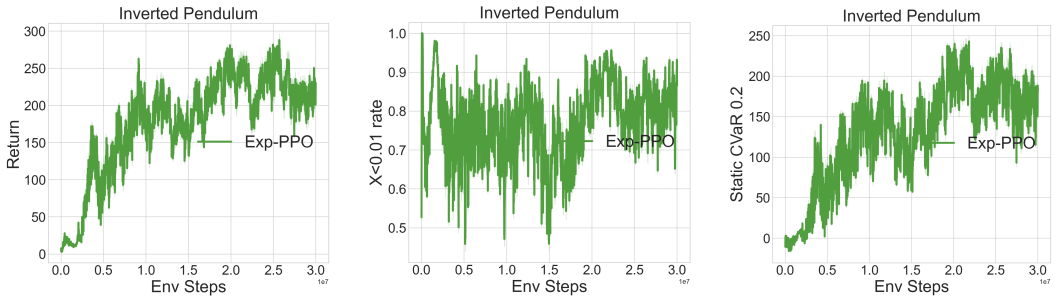
(d) CVaR 0.2 of return in Pendulum

Figure 7: Expected return and CVaR 0.2 of return in Inverted Pendulum under different risk levels. Curves are averaged over 10 seeds with shaded regions indicating standard errors.

Inverted Pendulum. The expected return and static 0.2-CVaR of the return under different risk levels of Exp-AC and CVaR-AC are shown in Fig. 7. For Exp-AC, when $\alpha = 0.4$, π lr=5e-5, Q lr=2.5e-4; when $\alpha = 0.3$, π lr=5e-5, Q lr=2.5e-4. For CVaR-AC, when $\alpha = 0.9$, π lr=5e-5, Q lr=1e-4, q lr=5e-4; when $\alpha = 0.5$, π lr=1e-4, Q lr=2e-4, q lr=1e-3.

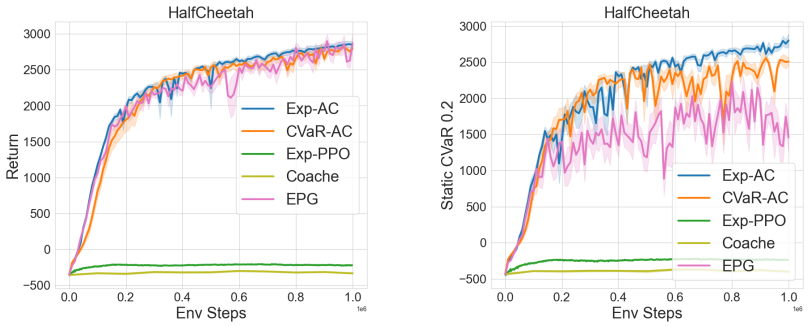
E.3.2 Longer environment steps of Exp-PPO in Inverted Pendulum

We observe that in Inverted Pendulum, Exp-PPO gains returns with longer environment steps. The learning curves are shown in Fig. 8.



(a) Expected return in Pendulum (b) Risk-averse rate in Pendulum (c) CVaR 0.2 of return in Pendulum

Figure 8: Expected return, risk-averse rate, and CVaR 0.2 of return in Inverted Pendulum. Curves are averaged over 10 seeds with shaded regions indicating standard errors.



(a) Expected return in HalfCheetah (b) CVaR 0.2 of return in HalfCheetah

Figure 9: Expected return and CVaR 0.2 of return in HalfCheetah. Curves are averaged over 10 seeds with shaded regions indicating standard errors.

E.3.3 HALFCHEETAH

We adopt HalfCheetah-v4 from Robust Gym (Gu et al., 2025) using an observation disruptor (i.e., adding noisy perturbation to observation). This disruptor is set using its API with `noise_factor = "state"`, `noise_type = "gauss"`, `noise_mu = 0.0`, and `noise_sigma = 0.05`. The maximum episode length is 500. Note that this domain does not provide a clear characterization of risk-averse behavior, while we can only observe the tail performance (e.g. 0.2 CVaR) of policies. The expected return and 0.2 CVaR of return curves (by sampling 15 trajectories for evaluation) are shown in Fig. 9. The parameter search range is the same as the deep learning cases above. Batch size is 256 for Exp-AC, CVaR-AC, and EPG (and they sample 10 actions for computing the policy gradient). Exp-PPO collects 2000 steps before updating policy. The chosen parameters are summarized in Table 5. Exp-AC and CVaR-AC achieves similar expected return as EPG while exhibit a better tail performance. We fail to get a good performing policy within the same parameter search range for Exp-PPO and method in Coache et al. (2023) in this domain.

Table 5: Learning parameters in HalfCheetah.

Method	α	π lr	Q (or V) lr	q lr
Exp-AC	0.4	3e-4	6e-4	-
CVaR-AC	0.5	3e-4	3e-4	1.5e-3
Exp-PPO	0.4	1e-4	1e-4	-
Coache	0.9	1e-4	2e-4	5e-4
EPG	-	3e-4	6e-4	-