EPISTEMIC ROBUST OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning learns policies from fixed datasets without further environment interaction. A key challenge in this setting is epistemic uncertainty, arising from limited or biased data coverage, particularly when the behavior policy systematically avoids certain actions. This can lead to inaccurate value estimates and unreliable generalization. Ensemble-based methods like SAC-N mitigate this by conservatively estimating Q-values using the ensemble minimum, but they require large ensembles and often conflate epistemic with aleatoric uncertainty. To address these limitations, we propose a unified and generalizable framework that replaces discrete ensembles with compact uncertainty sets over Q-values. We also introduce a benchmark for evaluating offline RL algorithms under risk-sensitive behavior policies, and demonstrate that our method achieves improved robustness and generalization over ensemble-based baselines across both tabular and continuous state domains.

Offline Reinforcement Learning (RL) seeks to learn policies from static datasets without further environment interaction. A key challenge is epistemic uncertainty arising from poor state-action coverage leading to unreliable value estimates and unsafe extrapolation, especially in domains where data collection is expensive or risky (e.g., healthcare, industrial control) Ghosh et al. (2022); Levine et al. (2020). Standard RL algorithms may overgeneralize in these regions, leading to unreliable value estimates and poor policy performance Yang et al. (2021). Ensemble-based methods like SAC-N address this by training multiple Q-networks and using a conservative Bellman target based on the pointwise minimum:

$$y(s,a) := r + \gamma \min_{i\in[N]} Q_\theta^{(i)}(s',a') - \alpha \log \pi_\phi(a'|s') \tag{1}$$

where $(s,a,r,s') \sim \mathcal{D}$ is a sample from the offline dataset, and $a' \sim \pi_\phi(\cdot|s')$ is drawn from the stochastic policy π_ϕ, parameterized by ϕ, $\gamma \in (0,1]$ is the discount factor and $\alpha > 0$ governs the entropy regularization.

The ensemble based formulation treats the minimum as a proxy for a lower confidence bound, encouraging conservative value estimates in uncertain regions. While effective, this method has limitations. Large ensemble sizes ($N \gg 1$) are often needed for reliable uncertainty estimates, increasing computational and memory costs Wen et al. (2020). The minimum also ignores inter-action correlations, limiting expressivity. Moreover, ensembles often conflate epistemic and aleatoric uncertainty Amini et al. (2020); Osband et al. (2023), making it difficult to distinguish model uncertainty from environment stochasticity, hindering robust and safe decision-making.

Epistemic uncertainty can persist even with large datasets when the behavior policy is biased. In the machine replacement problem Wiesemann et al. (2013), where an agent decides whether to continue operating or replace a degrading machine across 10 states, a risk-averse policy may replace early to avoid failure, while a risk-seeking one may delay to reduce cost. These choices induce systematically different state-action coverage, leading to high epistemic uncertainty in underexplored regions Schweighofer et al. (2022). This issue is especially pronounced in offline RL, where no further interaction is possible to resolve uncertainty. Example discussed in Appendix illustrates this with optimal and behavioral policies under different risk tolerances and the resulting coverage distributions.

To overcome these issues, we propose replacing the discrete ensemble $\{Q^{(i)}(s,a)\}_{i=1}^N$ with a compact uncertainty set $\mathcal{U}(s) \subset \mathbb{R}^{|\mathcal{A}|}$ defined per state. This yields a set-based Bellman target:

$$y(s,a) := r + \gamma \min_{\mathbf{q}\in\mathcal{U}(s')} \mathbb{E}_{a'\sim\pi_\phi(\cdot|s')}[q(a') - \alpha \log \pi_\phi(a'|s')], \tag{2}$$

where $\mathcal{U}(s')$ represents plausible Q-value vectors over actions at state s'. This formulation enables richer and more structured modeling of epistemic uncertainty, with improved sample efficiency and robustness.

Our contributions:

- We introduce ERSAC, a generalization of SAC-N using uncertainty sets to model structured epistemic uncertainty over Q-values.

- We integrate epistemic neural networks (Epinets) Osband et al. (2023) into ERSAC to directly produce uncertainty sets, removing the need for resampling.
- We develop a benchmark to evaluate offline RL under risk-sensitive behavior, demonstrating ERSAC's improved robustness and generalization across tasks.

For brevity, a detailed survey of related literature is deferred to Appendix .

PRELIMINARIES

We consider a Markov Decision Process (MDP) characterized by a possibly continuous state space $\mathcal{S}$, a discrete action space $\mathcal{A}$, a state-transition distribution $p(s_{t+1}|s_t, a_t)$, a reward function $r(s_t, a_t)$, and a discount factor $\gamma \in (0, 1)$. The reinforcement learning objective is to identify an optimal policy $\pi^*(\cdot|s)$, with $\pi^*(a|s)$ defining the likelihood of doing action a when in state s, that maximizes the expected discounted cumulative reward $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$. Below, we summarize the Soft Actor-Critic (SAC) Algorithm and one of its adaptations for offline RL that performs conservative updates using an ensemble of Q-functions.

SOFT ACTOR CRITIC (SAC)

The SAC framework optimizes the objective,

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right) \right],$$

where $\mathcal{H}(\pi(\cdot|s)) = -\sum_{a \in \mathcal{A}} \pi(a|s) \log \pi(a|s)$ is the entropy of the policy, and α controls the trade-off between exploration and exploitation.

SAC employs parametric approximations for both the Q-function $Q_\theta(s, a)$ and the policy $\pi_\phi(a|s)$, which are updated using off-policy data from a replay buffer. The Q-function minimizes temporal-difference error, while the policy is optimized to maximize expected entropy-regularized Q-values, $\mathbb{E}_{a \sim \pi_\phi(\cdot|s)} \left[Q_\theta(s, a) - \alpha \log \pi_\phi(a|s)\right]$. In this work, we use a discrete-action variant of SAC introduced in Christodoulou (2019), and refer the reader to their work for implementation and theoretical details.

SAC WITH AN ENSEMBLE OF Q-FUNCTIONS (SAC-N)

While SAC provides a stable framework for policy learning, applying it to offline RL is challenging since the agent relies solely on a fixed dataset. This makes SAC susceptible to overestimation bias, where the Q-function extrapolates inaccurately to out-of-distribution state-action pairs. Such bias is problematic during policy improvement, which favors actions with high Q-values, potentially leading to unsafe or suboptimal behavior. To mitigate this, An et al. (2021) proposed SAC-N, which uses an ensemble of N Q-functions $\{Q_{\theta_i}\}_{i=1}^{N}$ to capture epistemic uncertainty and reduce overestimation. Each $Q\theta_i$ estimates expected return, and a target ensemble $\{Q_{\theta'_i}\}_{i=1}^{N}$ is updated via Polyak averaging. The Q-function update adopts a clipped double Q-learning–style target (Fujimoto et al., 2018), extended in SAC-N by taking the minimum over the ensemble:

$$y(r, s', a') := \; r + \gamma \left(\min_i Q_{\theta'_i}(s', a') - \alpha \log \pi_\phi(a' \mid s') \right) \tag{3}$$

Using the minimum over the ensemble provides a conservative estimate of the expected return, reducing propagation of overestimated values from out-of-distribution state-action pairs common in offline datasets. Each Q-function Q_{θ_i} is updated by minimizing the mean squared Bellman error between its prediction and the target $y(r, s', a')$:

$$\mathcal{L}_Q(\theta_i) := \; \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D}, \\ a' \sim \pi_\phi(\cdot|s')}} \left[\Big(Q_{\theta_i}(s, a) - y(r, s', a') \Big)^2 \right] \tag{4}$$

where $\mathcal{D}$ denotes the static replay buffer of environment interactions, which, unlike in online RL, is fixed and is collected a priori without further interactions. The policy π_ϕ is then optimized to maximize the conservative estimate of the expected return (minimum Q-value across the ensemble) while incorporating the entropy regularization term:

$$\mathcal{J}_\pi(\phi) := \mathbb{E}_{\substack{s \sim \mathcal{D}, \\ a \sim \pi_\phi(\cdot|s)}} \left[\min_i Q_{\theta_i}(s, a) - \alpha \log \pi_\phi(a \mid s) \right] \tag{5}$$

This objective balances maximizing a conservative estimate of expected returns with encouraging high entropy, which promotes stochastic action selection. Greater entropy helps the policy explore beyond frequent actions in the offline dataset, particularly useful early in training to avoid overfitting to spurious correlations. Following Haarnoja et al. (2018), the entropy coefficient α is learned by minimizing a dual objective that aligns policy entropy with a target value, allowing the agent to maintain high entropy under uncertainty and gradually shift toward reward maximization.

Although SAC-N mitigates overestimation by maintaining an ensemble of Q-functions, it often requires a large ensemble size for stable performance. To address this, An et al. (2021) introduced the **Ensemble-Diversified Actor-Critic (EDAC)**, which adds a diversification term to encourage diversity among the Q-function ensemble members. In continuous action setting, they quantify similarity using an ensemble similarity (ES) metric defined as:

$$\frac{\langle \nabla_a Q_{\theta_i}(s,a), \nabla_a Q_{\theta_j}(s,a) \rangle}{\|\nabla_a Q_{\theta_i}(s,a)\| \|\nabla_a Q_{\theta_j}(s,a)\|},$$

which measures the cosine similarity between the gradients of different Q-functions with respect to the action vector. In the discrete action setting, where $\nabla_a Q(s,a)$ is ill defined, we adapt the ES metric by instead computing the mean squared deviation between the Q-values across all actions. Specifically, we define $g_\theta(s,a) := \big(Q_\theta(s,a') - Q_\theta(s,a)\big)_{a' \in \mathcal{A}}$, and compute the cosine similarity between $g_{\theta_i}(s,a)$ and $g_{\theta_j}(s,a)$:

$$\mathrm{ES}_{\theta_i,\theta_j}(s,a) := \frac{\sum_{a' \in \mathcal{A}} \big(Q_{\theta_i}(s,a') - Q_{\theta_i}(s,a)\big)\big(Q_{\theta_j}(s,a') - Q_{\theta_j}(s,a)\big)}{\sqrt{\sum_{a' \in \mathcal{A}} \big(Q_{\theta_i}(s,a') - Q_{\theta_i}(s,a)\big)^2}\sqrt{\sum_{a' \in \mathcal{A}} \big(Q_{\theta_j}(s,a') - Q_{\theta_j}(s,a)\big)^2}}.$$

The diversification loss is then given by:

$$\mathcal{L}_{\mathrm{ES}}(\theta) := \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\sum_{i=1}^{N}\sum_{j=i+1}^{N} \mathrm{ES}_{\theta_i,\theta_j}(s,a)\right].$$

where θ is short for $\{\theta_i\}_{i=1}^N$. The overall loss for each Q-function incorporates this diversification term:

$$\bar{\mathcal{L}}_Q(\theta) := (1/N)\sum_{i=1}^{N} \mathcal{L}_Q(\theta_i) + \eta \mathcal{L}_{\mathrm{ES}}(\theta), \tag{6}$$

where η is a hyperparameter controlling the strength of the diversity regularization. Encouraging diversity among the Q-functions was shown empirically to improve uncertainty estimation and leads to more reliable policy learning.

Epistemic Robustness with SAC

We start by formalizing the uncertainty captured by such an ensemble by modeling the long term actions values at a given state s as a distribution $F_\theta^q(s) \in \mathcal{M}(\mathbb{R}^{|\mathcal{A}|})$. Here, $F_\theta^q(s)$ defines a probability measure over Q-value vectors $q \in \mathbb{R}^{|\mathcal{A}|}$, induced by the variability among the Q-functions, and parameterized through θ. Each sample $\tilde{q} \sim F_\theta^q(s)$ is a vector in $\mathbb{R}^{|\mathcal{A}|}$ representing the epistemic uncertainty about the action-wise values $Q(s,\cdot)$. For example, in the case of SAC-N, this distribution takes the form of a scenario-based distribution:

$$F_\theta^q(s) := \frac{1}{N}\sum_{i=1}^{N} \delta_{Q_{\theta_i}(s,\cdot)}, \tag{7}$$

where δ_x is the Dirac measure centered at $x \in \mathbb{R}^{|\mathcal{A}|}$. Given a Q-value distribution $F_\theta^q : \mathcal{S} \to \mathcal{M}(\mathbb{R}^{|\mathcal{A}|})$, mapping each state $s \in \mathcal{S}$ to a probability measure over Q-value vectors, we define an uncertainty set operator,

$$\mathcal{U} : \mathcal{M}(\mathbb{R}^{|\mathcal{A}|}) \to \mathcal{C}(\mathbb{R}^{|\mathcal{A}|}),$$

that maps a Q-value distribution to a compact set of plausible Q-value vectors. The composition $\mathcal{U} \circ F_\theta^q : \mathcal{S} \to \mathcal{C}(\mathbb{R}^{|\mathcal{A}|})$ defines an epistemic uncertainty set $\mathcal{U}(F_\theta^q(s))$ in each state s, which can be used to construct robust evaluation and optimization of policies. For notational simplicity, we will use $\mathcal{U}_\theta(s)$ as shorthand for $\mathcal{U}(F_\theta^q(s))$ when the dependencies on F_θ^q are clear from context.

In the next section, we introduce our proposed framework, **Epistemic Robust Soft Actor-Critic (ERSAC)**, which generalizes SAC-N by leveraging uncertainty sets derived from Q-value distributions. We first present an ensemble-based version of ERSAC and highlight its connection to SAC-N. We then formalize the algorithm, detailing its key components, the set-based Bellman backup and the robust policy update.

The Epistemic Robust SAC (ERSAC) Model

As in SAC-N, ERSAC trains the Q-function by minimizing the expected squared Bellman error between a sampled realization and a conservative target derived from the Q-distribution F^q_θ. Specifically, for each next state $s' \in \mathcal{S}$, the target in (3) is modified to:

$$y(r, s') := \; r + \gamma \left(\min_{q \in \mathcal{U}(F^q_{\theta'}(s'))} \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} \left[q(s', a') - \alpha \log \pi_\phi(a' \mid s') \right] \right) \tag{8}$$

where the minimum operator provides a robust estimate of the regularized expected total discounted return. We refer the reader to Ben-Tal et al. (2015) for closed form expressions of $\min_{q \in \mathcal{U}} \langle v, q \rangle$ for a list of popular forms of uncertainty sets. The loss function in (4) is then redefined as:

$$\mathcal{L}^R_Q(\theta) := \mathbb{E}_{(s,a,r,s') \sim \mathcal{D},\; \tilde{q} \sim F^q_\theta(s)} \left[\left(\tilde{q}(a) - y(r, s') \right)^2 \right]. \tag{9}$$ [1]

Similar to the Q-value target, the policy loss in the epistemic robust setting replaces the ensemble minimum with a worst-case expectation over the uncertainty set. The robust policy loss (5) becomes:

$$\begin{aligned} \mathcal{J}^R_\pi(\phi) : &= \mathbb{E}_{s \sim \mathcal{D}} \left[\min_{q \in \mathcal{U}_\theta(s)} \mathbb{E}_{a \sim \pi_\phi(\cdot|s)} \left[q(a) - \alpha \log \pi_\phi(a \mid s) \right] \right] \\ &= \mathbb{E}_{\substack{s \sim \mathcal{D}, \\ a \sim \pi_\phi(\cdot|s)}} \left[\min_{q \in \mathcal{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q \rangle - \alpha \log \pi_\phi(a \mid s) \right] \end{aligned} \tag{10}$$

Importantly, when using an ensemble based representation, the ERSAC formulation encompasses SAC-N as a special case under a particular choice of uncertainty set. We formalize this connection in the following proposition and defer the proof to Appendix .

Proposition 1. *Let $F^q_\theta(s)$ be defined as in Equation (7), and let the uncertainty set operator be defined as*

$$\mathcal{U}_{box}(F^q_\theta(s)) := \bigtimes_{a \in \mathcal{A}} \left[\underset{\tilde{q} \sim F^q_\theta(s)}{essinf} \, [\tilde{q}(a)], \;\; \underset{\tilde{q} \sim F^q_\theta(s)}{esssup} \, [\tilde{q}(a)] \right] \tag{11}$$

i.e., a coordinate-wise box containing the support of $F^q_\theta(s)$, under which the robust losses reduce to those of SAC-N: $\mathcal{L}^R_Q(\theta) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_Q(\theta_i) + C$ and $\mathcal{J}^R_\pi = \mathcal{J}_\pi$, for some constant $C \in \mathbb{R}$ independent of θ.

This result demonstrates that ERSAC generalizes SAC-N under a unified uncertainty set framework. In the next section, for an arbitrary compact set representation $\mathcal{U}_\theta(s)$, we outline the detailed training algorithm.

The ERSAC Training Algorithm

Previously, we modeled $F^q_\theta(s)$ such that each sample $\tilde{q} \sim F^q_\theta(s)$ is a Q-value vector in $\mathbb{R}^{|\mathcal{A}|}$, representing $Q(s, \cdot)$. To generalize this, we adopt the reparameterized formulation from Assumption 2.

Assumption 2. F^q_θ is associated to a sampling operator $\mathfrak{q}_\theta(s, a, z)$ and a distribution $F_z \in \mathcal{M}(\mathbb{R}^{d_z})$, such that $\mathfrak{q}_\theta(s, \cdot, \tilde{z})$ follows $F^q_\theta(s)$ when $\tilde{z} \sim F_z$.

Given a noise sample $\tilde{z} \sim F_z$, a corresponding Q-vector sample $\tilde{q} \sim F^q_\theta(s)$ is obtained by evaluating the sampling operator over all actions:

$$\tilde{q}(a) := \mathfrak{q}_\theta(s, a, \tilde{z}), \quad \text{for all } a \in \mathcal{A}.$$

This reparameterization generalizes the ensemble model in Equation 7 as a special case, where the latent variable $\tilde{z} \in \{1, \ldots, N\}$ indexes a finite set of Q-functions, and $q_\theta(s, a, \hat{z}) = Q_{\theta_{\tilde{z}}}(s, a)$.

In order to minimize $\mathcal{L}^R_Q$, when Assumption 2 is satisfied, one can use a popular reparametrization trick to derive a gradient for the critic parameters θ as:

$$\begin{aligned} \nabla_\theta \mathcal{L}^R_Q(\theta) &= \nabla_\theta \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ \tilde{z} \sim F_z}} \left[\left(\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s') \right)^2 \right] \\ &= \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ \tilde{z} \sim F_z}} \left[2 \left(\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s') \right) \nabla_\theta \mathfrak{q}_\theta(s, a, \tilde{z}) \right] \end{aligned}$$

[1] It is important to note that without additional regularization, the objective in (9) may admit a degenerate solution $F^q_{\theta^*}(s) = \delta_{\bar{q}(s,\cdot)}$, where $\bar{q}(s, a) := \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}[y(r, s')]$, which collapses the distribution to a deterministic point estimate. In practice, this requires regularization strategies such as early stopping, entropy constraints on F^q_θ, or prior-based regularization to avoid mode collapse.

This gives rise to the stochastic update $\theta \;\leftarrow\; \theta - \eta_Q\, 2\big(\mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s')\big)\nabla_\theta \mathfrak{q}_\theta(s,a,\tilde{z})$. Optimizing $\mathcal{J}_\pi^R$ is a bit more complex; we begin by letting $q^*(s,\cdot;\phi)$ denote any statewise adversarial Q-value vector for policy π_ϕ:

$$q^*(s,\cdot\,;\phi) \in \arg\min_{q\in\mathcal{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q\rangle,\; \forall s \in \mathcal{S}, \tag{12}$$

which is well-defined due to compactness of $\mathcal{U}_\theta(s)$. Then, noting that the function

$$\begin{aligned} f(\pi) :=\; & \mathbb{E}_{s\sim\mathcal{D},a\sim\pi(\cdot|s)}\left[\min_{q\in\mathcal{U}_\theta(s)} \langle \pi(\cdot \mid s), q\rangle - \alpha \log \pi(a \mid s)\right] \\ =& \mathbb{E}_{s\sim\mathcal{D}}\left[\min_{q\in\mathcal{U}_\theta(s)} \langle \pi(\cdot\,|\,s), q\rangle - \alpha \mathbb{E}_{a\sim\pi(\cdot|s)}[\log \pi(a\,|\,s)]\right] \end{aligned}$$

is concave with respect to π, one can invoke the envelope theorem to identify one of its supergradients as

$$\nabla_\pi \mathbb{E}_{s\sim\mathcal{D}}\Big[\langle \pi(\cdot \mid s),\, q^*(s,\cdot\,;\phi)\rangle - \alpha\, \mathbb{E}_{a\sim\pi(\cdot|s)}[\log \pi(a \mid s)]\Big] \in \nabla_\pi f(\pi)$$

We therefore obtain, fixing $\bar{\phi}$ to ϕ that:

$$\nabla_\phi \mathcal{J}_\pi^R(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[\sum_{a\in\mathcal{A}} q^*(s,a\,;\phi)\, \nabla_\phi \pi_\phi(a \mid s) - \alpha\, \nabla_\phi \langle \pi_\phi(\cdot \mid s),\, \log \pi_\phi(\cdot \mid s)\rangle\right] \tag{13}$$

This produces a standard entropy-regularized policy gradient, but is evaluated with respect to the worst-case value vector $q^*(s,\cdot\,;\phi)$ in the uncertainty set, providing robustness to epistemic uncertainty. We summarize the training procedure for Robust SAC-N in Algorithm 1 in Appendix .

SAMPLE-BASED CONSTRUCTION OF $\mathcal{U}_\theta(s)$ FROM $\mathfrak{q}_\theta(s,a,\tilde{z})$

In practice, one often approximates $F_\theta^q(s)$ using Monte Carlo samples, which form an empirical distribution $\widehat{F}_\theta^q(s)$. Having access to $\widehat{F}_\theta^q(s)$, one can approximate $\mathcal{U}(F_\theta^q(s))$ with $\mathcal{U}(\widehat{F}_\theta^q(s))$. Different choices of $\mathcal{U}(\widehat{F}_\theta^q(s))$ lead to varying trade-offs between computational tractability, policy sensitivity, and expressiveness. In the remainder of this section, we present three popular sets from the literature of robust optimization: box set, convex hull set and ellipsoidal set.

Box set: Let $\{\tilde{z}_i\}_{i=1}^N$ be N values sampled from F_z. The simplest construction is the box set introduced in (11), which defines $\mathcal{U}_\theta(s)$ as the Cartesian product of the intervals covering $\tilde{q}(a)$ for each action. In a sample-based setting, this reduces to :

$$\mathcal{U}_{\text{box}}(\widehat{F}_\theta^q(s)) := \bigtimes_{a\in\mathcal{A}}\left[\min_{i=1,\ldots,N} \mathfrak{q}_\theta(s,a,\tilde{z}_i),\, \max_{i=1,\ldots,N} \mathfrak{q}_\theta(s,a,\tilde{z}_i)\right] \tag{14}$$

Convex Hull Set: A more expressive alternative is the uncertainty set operator that produces the convex hull of the support of $F_\theta^q(s)$. In a sample-based setting, this reduces to:

$$\mathcal{U}_{\text{hull}}(\widehat{F}_\theta^q(s)) := \left\{\sum_{i=1}^N \lambda_i\, \mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) \;\middle|\; \exists\, \lambda \in \mathbb{R}^N,\; \lambda_i \geq 0\; \forall i, \sum_{i=1}^N \lambda_i = 1\right\} \tag{15}$$

The worst-case Q-vector is $q^*(s,a;\phi) = \mathfrak{q}_\theta(s,a,z^*(s,\phi))$, where $z^*(s,\phi) \in \arg\min_i \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[\mathfrak{q}_\theta(s,a,\tilde{z}_i)]$.

Ellipsoidal Set: In this work, we will mainly consider an ellipsoidal set operator that aim to cover a certain proportion υ of the total mass of $F_\theta^q(s)$. In a sample-based setting, this can be done by estimating the empirical mean and covariance of the sampled Q-vectors:

$$\hat{\mu}(s) := \tfrac{1}{N}\sum_{i=1}^N \mathfrak{q}_\theta(s,\cdot,\tilde{z}_i), \qquad \widehat{\Sigma}(s) := \tfrac{1}{N}\sum_{i=1}^N \big(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s)\big)\big(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s)\big)^\top$$

and estimating the radius as

$$\widehat{\Upsilon}(s) := \inf\left\{\Upsilon \;\middle|\; \frac{1}{N}\sum_{i=1}^N \mathbf{1}\left\{\big(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s)\big)^\top \widehat{\Sigma}(s)^{-1} \cdot \big(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s)\big) \leq \Upsilon^2\right\} \geq \upsilon\right\}$$

The corresponding uncertainty set is defined as:

$$\mathcal{U}_{\text{ell}}(\widehat{F}^q_\theta(s)) := \left\{ q \in \mathbb{R}^{|\mathcal{A}|} \,\middle|\, (q - \hat{\mu}(s))^\top \, \widehat{\Sigma}(s)^{-1} \cdot (q - \hat{\mu}(s)) \leq \widehat{\Upsilon}(s)^2 \right\} \tag{16}$$

This set encodes second-order structure and supports efficient optimization. When $\widehat{\Sigma}(s)$ is positive definite, the worst-case Q-vector under a given policy admits the closed-form solution:

$$q^*(s, \cdot\,; \phi) = \hat{\mu}(s) - \widehat{\Upsilon}(s) \cdot \frac{\widehat{\Sigma}(s)\pi_\phi(\cdot \mid s)}{\|\widehat{\Sigma}(s)^{1/2}\pi_\phi(\cdot \mid s)\|}.$$

For completeness, the detailed derivations of the policy-sensitive worst-case Q-vector under both the convex hull and ellipsoidal sets are provided in Appendix .

We refer the reader to Appendix for the pseudocode of the training algorithm based on box, convex hull (Algorithm 2) and ellipsoidal (Algorithm 3) uncertainty sets. A deeper discussion on how the choice of uncertainty set affects the sensitivity of the worst-case Q-vector to the policy π_ϕ, based on the Machine Replacement example introduced earlier, is provided in Appendix .

The ERSAC model with Epinet (ERSAC(Epi))

Recall from Assumption 2 that we require a parametric sampling operator $\mathfrak{q}_\theta(s, a, z)$, with $z \sim F_z$, such that $\mathfrak{q}_\theta(s, \cdot, z) \sim F^q_\theta(s)$, where $F^q_\theta(s) \in \mathcal{M}(\mathbb{R}^{|\mathcal{A}|})$ denotes a distribution over Q-value vectors. We instantiate this generative model using an Epistemic Neural Network (Epinet) introduced by Osband et al. (2023), which enables structured and differentiable sampling from a single neural network. An Epinet supplements a base network $\mu_{\theta_\mu}(s, a) \in \mathbb{R}$, parameterized by θ_μ, which yields the mean Q-value vector. From this base, we extract a feature representation $\psi_{\theta_\mu}(s) \in \mathbb{R}^{d_\psi}$, typically taken from the last hidden layer. Epistemic variation is introduced via a latent index $z \sim \mathcal{N}(0, I) \in \mathbb{R}^{d_z}$. These components are combined through a stochastic head $\sigma_{\theta_\sigma}(\psi_{\theta_\mu}(s), a, z) \in \mathbb{R}$, which modulates the structured uncertainty. The sampling operator for the Q-value vector is then defined as $\mathfrak{q}_\theta(s, \cdot, z) := \mu_{\theta_\mu}(s, \cdot) + \sigma_{\theta_\sigma}(\psi_{\theta_\mu}(s), \cdot, z),$. The stochastic head is constructed as $\sigma_{\theta_\sigma}(\psi, \cdot, z) := \sigma^{\text{L}}_{\theta_\sigma}(\psi, \cdot, z) + \sigma^{\text{P}}(\psi, \cdot, z)$ with $\sigma^{\text{L}}_{\theta_\sigma} : \mathbb{R}^{d_\psi} \times \mathcal{A} \times \mathbb{R}^{d_z} \to \mathbb{R}$ as a learnable function and $\sigma^{\text{P}} : \mathbb{R}^{d_\psi} \times \mathcal{A} \times \mathbb{R}^{d_z} \to \mathbb{R}$ as a fixed prior. The fixed prior network σ^{P} encodes initial epistemic uncertainty by inducing variability in predictions across samples of indices z. In well explored regions, $\sigma^{\text{L}}_{\theta_\sigma}$ can learn better distributions for the predictive uncertainty, while in data sparse areas, σ^{P} can induce the prior beliefs of the decision maker to guide conservative predictions. We can now use it to generate the realizations of the Q-value vectors at a given state s by drawing $z \sim \mathcal{N}(0, I)$ to form the empirical distribution $\widehat{F}_\theta(s)$ over Q values. This enables us to employ the sample based epistemic uncertainty sets introduced in the earlier section.

This construction yields a parameter efficient and fully differentiable reparameterization of the Q distribution. Further, one can train these networks using a perturbed squared loss inspired by Gaussian bootstrapping following the loss:

$$\mathcal{L}_Q^{ENN}(\theta) := \mathbb{E}_{(s,a,r,s',c)\sim\bar{\mathcal{D}},\ \tilde{z}\sim F_z}\left[\left(\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s') - \bar{\sigma}\langle c, \tilde{z}\rangle\right)^2\right] + \lambda_\mu\|\theta_\mu\|^2 + \lambda_\sigma\|\theta_\sigma\|^2$$

where each member (s, a, r, s') from the dataset $\mathcal{D}$ is augmented with some c randomly sampled from the surface of the unit sphere $\mathbb{S}^{d_z}$ to produce $\bar{\mathcal{D}}$, where $\bar{\sigma} > 0$ denotes the bootstrap noise scale, and where $\lambda_\zeta, \lambda_\eta$ are regularization coefficients. This loss encourages the network to match bootstrapped Q-targets while introducing variability across z samples. It can be minimized via standard stochastic gradient methods. The ENN critic updates thus become:

$$\theta_\mu \leftarrow\ \theta_\mu - 2\eta_Q \cdot \left(\tfrac{1}{|\bar{\mathcal{B}}|} \sum_{(s,a,r,s',c)\in\bar{\mathcal{B}}} \mathbb{E}_{\tilde{z}\sim F_z}\left[\left(\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s') - \bar{\sigma}\langle c, \tilde{z}\rangle\right)\right] \cdot \nabla_{\theta_\mu}\mu_{\theta_\mu}(s, a) + 2\lambda_\mu\theta_\mu\right) \tag{17}$$

$$\theta_\sigma \leftarrow\ \theta_\sigma - 2\eta_Q \cdot \left(\tfrac{1}{|\bar{\mathcal{B}}|} \sum_{(s,a,r,s',c)\in\bar{\mathcal{B}}} \mathbb{E}_{\tilde{z}\sim F_z}\left[\left(\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s') - \bar{\sigma}\langle c, \tilde{z}\rangle\right)\right] \cdot \nabla_{\theta_\sigma}\sigma^L_{\theta_\sigma}(\psi_{\theta_\mu}(s), a, \tilde{z}) + 2\lambda_\sigma\theta_\sigma\right) \tag{18}$$

To accelerate the evaluation of $\mathcal{U}(F^q_\theta(s)$ when using an ellipsoidal uncertainty set operator, we introduce additional structure in $\sigma^{\text{L}}_{\theta_\sigma}(\psi, \cdot, z)$ and $\sigma^{\text{P}}(\psi, \cdot, z)$ as outlined in Assumption 3, namely that both operators are linear in z.

Assumption 3. The stochastic heads $\sigma^{\mathrm{L}}_{\theta_\sigma}(\psi, a, z)$ and $\sigma^{\mathrm{P}}(\psi, a, z)$ are linear in z, i.e.,

$$\sigma^{\mathrm{L}}_{\theta_\sigma}(\psi, a, z) = \langle \bar{\sigma}^{\mathrm{L}}_{\theta_\sigma}(\psi, a), z\rangle, \quad \sigma^{\mathrm{P}}(\psi, a, z) = \langle \bar{\sigma}^{\mathrm{P}}(\psi, a), z\rangle,$$

for some mappings $\bar{\sigma}^{\mathrm{L}}_{\theta_\sigma} : \mathbb{R}^{d_\psi} \times \mathcal{A} \to \mathbb{R}^{d_z}$ and $\bar{\sigma}^{\mathrm{P}} : \mathbb{R}^{d_\psi} \times \mathcal{A} \to \mathbb{R}^{d_z}$.

Assumption 3 induces a Gaussian distribution,

$$\mathfrak{q}_\theta(s, \cdot, z) \sim \mathcal{N}(\mu_{\theta_\mu}(s), \Sigma_\theta(s)), \tag{19}$$

where the covariance is defined as, $[\Sigma_\theta(s)]_{a,a'} := \langle \bar{\sigma}^L_{\theta_\sigma}(\psi_{\theta_\mu}(s), a) + \bar{\sigma}^P(\psi_{\theta_\mu}(s), a), \bar{\sigma}^L_{\theta_\sigma}(\psi_{\theta_\mu}(s), a') + \bar{\sigma}^P(\psi_{\theta_\mu}(s), a')\rangle$. This gives rise to the Epinet based ellipsoidal set:

$$\mathcal{U}^{ENN}_{\mathrm{ell}}(s) := \left\{ q \in \mathbb{R}^{|\mathcal{A}|} \;\middle|\; (q - \mu_{\theta_\mu}(s))^\top \Sigma_\theta(s)^{-1} \cdot (q - \mu_{\theta_\mu}(s)) \le F^{-1}_{\chi^2_{|\mathcal{A}|}}(\upsilon) \right\} \tag{20}$$

Here, $F^{-1}_{\chi^2_{|\mathcal{A}|}}(\upsilon)$ denotes the inverse CDF of the χ^2 distribution with $|\mathcal{A}|$ degrees of freedom, yielding an efficient alternative to ensemble based uncertainty modeling with a closed form worst case Q-vector. The assumption of linear stochastic heads in Epinet is mainly for computational efficiency, allowing closed-form mean and covariance estimates for ellipsoidal uncertainty sets. While this may limit expressivity compared to nonlinear heads, it is generally sufficient for capturing epistemic uncertainty in many RL settings. In highly non-Gaussian cases, richer parameterizations or sampling-based approaches may be needed. Relaxing this assumption could enable more flexible uncertainty modeling, but at increased computational cost.

The training procedure for ERSAC with Epinet (ERSAC(Epi)) mirrors the ensemble based variant (Algorithm 3) but avoids sampling by leveraging the structured Epinet model. The mean and covariance are directly obtained as $\mu_{\theta_\mu}(s)$ and $\Sigma_\theta(s)$ from the deterministic and stochastic heads under Assumption 3. The ellipsoidal radius is set to $\Upsilon^2(s) = F^{-1}_{\chi^2_{|\mathcal{A}|}}(\upsilon)$, ensuring a υ-level confidence set. This enables efficient, fully differentiable updates for both the Bellman target and policy gradient. See Appendix , Algorithm 4 for full details.

Experiments

In this section, we present a comprehensive empirical evaluation of our framework for epistemic robustness in offline reinforcement learning. Epistemic uncertainty is captured via uncertainty sets that integrate seamlessly into robust policy optimization. The three sample-based uncertainty sets lead to three ERSAC variants: **SAC-N** (ie. ERSAC with a box set over N ensembles), **ERSAC-CH-N** (convex hull over ensembles), and **ERSAC-Ell-N** (ellipsoids from empirical mean and covariance). We also evaluate **ERSAC-Ell-Epi**, which replaces the ensemble with N samples from ERSAC-EPI to produce a sample-based ellipsoid. Lastly, **ERSAC-Ell-Epi*** leverages the structured stochastic head $\sigma_{\theta_\sigma}(\psi, \cdot, z)$ (see Assumption 3) to construct ellipsoidal sets directly, without sampling. The code can be found on GitHub[2].

Our experiments span a diverse set of environments, including tabular domains (Machine Replacement and Riverswim), classic control benchmarks (CartPole and LunarLander). Across these domains, we evaluate each method's ability to learn effective policies under distributional shifts arising due to changes in behavior policies and limited data coverage.

A key contribution of our work is a novel offline RL benchmarking framework that enables control over the risk sensitivity of the behavior policy used to generate offline datasets. By adjusting the level of optimism or pessimism through expectile-based value learning, we can systematically evaluate how the nature of behavioral data affects the performance of offline RL algorithms. To induce risk sensitivity, we employ a modified actor-critic algorithm incorporating the dynamic expectile risk measure (Marzban et al. (2023)). For each (s, a), critic target is computed using a bootstrapped expectile estimate:

$$y := \operatorname*{arg\,min}_{z \in \mathbb{R}} \sum_{j=1}^{M} \left| \mathbb{I}\left(z < r + \gamma \max_{a'} Q_\theta(s'_j, a') \right) - \tau \right| \cdot \left(z - r(s, a) - \gamma \max_{a'} Q_\theta(s'_j, a') \right)^2,$$

and the critic minimizes squared error to this target. The actor is trained via a standard policy gradient to maximize expected Q-values.

After a fixed number of training steps, the resulting policy π_ϕ reflects the desired level of risk sensitivity through τ. We then collect an offline dataset of size N using ε-greedy interaction with the environment, selecting random actions with probability $\varepsilon = 0.1$. This yields datasets with systematically varying behavioral bias. Full implementation details are provided in Appendix 5.

[2] https://anonymous.4open.science/r/ERSAC-5C0E/discrete_env_utils.py

Env	DS	SAC-N	CH-N	Ell-N	Ell_0.9-N	Beh. Policy
Machine Replacement	$10\times$	$\underline{84 \pm 3}$	86 ± 2	89 ± 2	$\mathbf{90 \pm 2}$	93 ± 2
	$100\times$	97 ± 2	$\mathbf{96 \pm 2}$	$\underline{94 \pm 2}$	95 ± 2	93 ± 2
	$1000\times$	97 ± 2	97 ± 2	$\underline{97 \pm 2}$	$\mathbf{97 \pm 1}$	93 ± 2
RiverSwim	$10\times$	$\underline{47 \pm 3}$	58 ± 3	55 ± 3	$\mathbf{60 \pm 3}$	5 ± 4
	$100\times$	$\underline{96 \pm 2}$	97 ± 2	97 ± 2	$\mathbf{98 \pm 2}$	5 ± 4
	$1000\times$	99 ± 1	99 ± 1	100 ± 0	$\mathbf{100 \pm 0}$	5 ± 4

(a) Tabular environments

Env	DS	SAC-N	CH-N	Ell_0.9-N	Ell-Epi	Ell-Epi*	Beh. Policy
CartPole	1k	$\underline{76 \pm 3}$	74 ± 2	$\mathbf{79 \pm 2}$	79 ± 2	77 ± 2	90 ± 2
	10k	$\underline{96 \pm 2}$	98 ± 1	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	90 ± 2
	100k	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	90 ± 2
LunarLander	1k	$\underline{69 \pm 2}$	74 ± 2	97 ± 2	97 ± 2	$\mathbf{97 \pm 2}$	89 ± 3
	10k	$\underline{93 \pm 2}$	99 ± 2	101 ± 1	100 ± 2	$\mathbf{102 \pm 1}$	89 ± 3
	100k	$\underline{98 \pm 2}$	100 ± 2	104 ± 1	$\mathbf{107 \pm 2}$	106 ± 1	89 ± 3

(b) Gym environments

Table 1: Returns aggregated across $\tau \in \{0.1, 0.5, 0.9\}$ for each dataset size. Bold indicates best method, underline the worst, when mean differences ≥ 1.

Evaluation on tabular tasks

We begin our evaluation with two tabular MDP environments: the *Machine Replacement* problem and *Riverswim*. These settings offer interpretable structure while capturing core challenges of offline RL, including sparse coverage and sensitivity to policy extrapolation. Crucially, the tabular setup isolates the role of epistemic uncertainty without confounding effects from deep RL (e.g., overfitting or instability), enabling a clean assessment of how different uncertainty set constructions mitigate overestimation. For each environment, we construct offline datasets by systematically varying two parameters: dataset size and behavior policy risk sensitivity. To assess sample efficiency, we vary dataset size as $10 \times |\mathcal{S}|$, $100 \times |\mathcal{S}|$, and $1000 \times |\mathcal{S}|$, where $|\mathcal{S}|$ is the number of states. These reflect increasing coverage, with diminishing returns observed beyond $1000 \times |\mathcal{S}|$ as empirical dynamics approximate the true model. To induce behavioral bias and modulate epistemic uncertainty, we vary the behavior policy using the dynamic expectile risk measure at three levels: risk-seeking ($\tau = 0.1$), risk-neutral ($\tau = 0.5$), and risk-averse ($\tau = 0.9$). These settings lead to distinct exploration behaviors and result in datasets with varying state-action coverage.

We evaluate performance using normalized returns, which measure the improvement of a learned policy over a uniformly random policy, scaled relative to the performance of the optimal policy. Specifically, for a learned policy π, we compare $(J(\pi) - J(\pi_{\text{rand}}))\,/\,(J(\pi^*) - J(\pi_{\text{rand}}))$ where $J(\pi)$ is the expected returns under policy π, computed as the average return over 100 evaluation episodes, π_{rand} is the random policy, and π^* is the optimal policy.

Table 1a summarizes normalized returns aggregated over τ values for each dataset size, while full results across all τ settings are provided in Table 4 in Appendix . In low-data settings (e.g., 100 samples), CH-N and Ell_0.9-N outperform B-N by up to 75%, highlighting the benefit of structured epistemic reasoning under sparse coverage. As dataset size increases, all methods improve, but structured sets consistently converge faster toward optimal returns. Under risk-averse regimes ($\tau = 0.9$), where epistemic uncertainty is highest, ellipsoidal variants maintain robustness, with Ell-N and Ell_0.9-N effectively modulating conservativeness to preserve performance.

A key strength of the ellipsoidal set is its tunable scaling parameter ϵ, which controls conservativeness. To assess its effect, we compare ellipsoids covering 100% (Ell-N) versus 90% (Ell_0.9-N) of ensemble samples. The tighter 90% set often outperforms, likely due to excluding outlier critics and avoiding over-pessimism. We therefore adopt 90% coverage as the default in subsequent Gym based experiments.

Evaluation on Gym environments

We next evaluate the proposed methods on two widely used Gym environments, *CartPole* and *LunarLander*. CartPole is a standard control task with binary rewards and continuous states, while LunarLander presents greater complexity with shaped rewards and a higher-dimensional state-action space. As in the tabular setting, we construct offline datasets by varying two factors: dataset size and behavior policy risk profile. For each environment, we generate nine datasets by crossing three dataset sizes (1K, 10K, and 100K transitions) with three expectile levels: $\tau = 0.1$ (risk-seeking), $\tau = 0.5$ (risk-neutral), and $\tau = 0.9$ (risk-averse). Behavior policies are trained to convergence using a dynamic expectile based actor-critic algorithm, and fixed trajectories are collected for each configuration.

Table 1b summarizes normalized returns aggregated over τ values for each dataset size, while full results across all τ settings are provided in Table 5 in Appendix . We consider the policy trained under the risk neutral behavior($\tau = 0.5$) as the reference optimal policy. First, models CH-N, Ell_0.9-N, Ell-Epi consistently outperform the box baseline B-N, particularly in data scarce and risk averse settings where epistemic uncertainty plays a larger role. When we aggregate returns across dataset sizes by risk level (As presented in Table 2), we observe that Ell_0.9-N consistently achieves strong performance under risk-neutral and risk-seeking behavior policies, suggesting that the method effectively leverages optimistic data to enhance policy learning.

Env	$\tau = 0.1$	$\tau = 0.5$	$\tau = 0.9$
CartPole	95 ± 8 (1)	93 ± 14 (2)	92 ± 14 (3)
LunarLander	103 ± 7 (1)	99 ± 5 (2)	99 ± 5 (2)
MR	93 ± 1 (3)	95 ± 4 (1)	94 ± 3 (2)
RS	87 ± 15 (2)	87 ± 17 (1)	84 ± 22 (3)

Table 2: Agg. performance of **Ell_0.9-N** across environments with mean ± std and within-environment rank (1 = best).

Ellipsoidal variants show strong, often best, performance across settings. Ell-Epi* matches or outperforms the ensemble based Ell_0.9-N in several cases, highlighting Epinet-based uncertainty as an efficient alternative. We observed that Ell-Epi* achieves comparable performance with significantly lower compute (see Appendix for details), making it attractive for scaling to complex domains.

To further understand how uncertainty sets affect learning dynamics, we analyze policy entropy during training. We observed that Box-based methods (B-N) maintain consistently lower entropy, indicating less stochastic and more prematurely deterministic policies. This often leads to suboptimal convergence. In contrast, **CH-N**, **Ell-N**, and **Ell-Epi** allow more flexible shaping of $q^*(s, \cdot; \phi)$, encouraging exploration and enabling better identification of high-reward actions under offline constraints. We refer the reader to Appendix for a detailed report.

Discussion

This work introduced Epistemic Robust Soft Actor-Critic (ERSAC), a unified framework for offline RL that robustly models epistemic uncertainty using structured uncertainty sets over Q-values. By replacing ensemble-based pessimism with sets such as box, convex hull, and ellipsoid, ERSAC enables conservative yet flexible value estimation and policy optimization. The framework generalizes SAC-N as a special case and supports multiple set constructions, each with trade-offs in expressiveness and computational cost. The Epinet-based variant further allows closed-form ellipsoidal uncertainty sets, reducing runtime without sacrificing performance.

A central aspect of our evaluation is the use of risk-aware behavior policies to systematically induce coverage bias in offline datasets. By varying the risk sensitivity of the data-generating policy, we provide a controlled and generalizable way to study the impact of behavior policy quality on offline RL. This approach allows for precise modulation of epistemic uncertainty, highlighting regimes where conservative value estimation is most critical. Our results show that ERSAC's structured uncertainty sets are particularly effective in settings with poor or biased coverage, such as risk-averse or highly optimistic policies, where epistemic uncertainty is highest. As coverage improves, uncertainty sets shrink and the performance of ERSAC approaches that of standard ensemble methods. This paradigm enables a nuanced understanding of robustness in offline RL and our benchmark for risk-sensitive behavior policies further underscores the value of this approach for systematic assessment.

Future directions include extending epistemic robustness to multi-agent and hierarchical reinforcement learning, integrating risk-aware objectives. Establishing finite-sample generalization guarantees and robust regret bounds under epistemic uncertainty remains an open challenge. In general, our results show that structured and efficient epistemic modeling provides a foundation for safe, generalizable, and scalable offline RL.

Reproducibility Statement

The paper provides detailed information to facilitate reproducibility. Formal definitions, assumptions, and proofs of the main theoretical results are included in the Appendix. Algorithmic details, including pseudocode for all proposed ERSAC variants, are presented in Appendix Sections on implementation and training. The offline data generation procedure, hyperparameter settings, and evaluation protocols are described in detail, with full experimental results and additional tables and figures provided in the Appendix. An anonymous GitHub repository containing the full implementation and benchmark framework is linked in the experiments section.

References

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937, 2020.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

Aharon Ben-Tal, Dick Den Hertog, and Jean-Philippe Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical programming*, 149(1):265–299, 2015.

Dimitris Bertsimas, Christopher McCord, and Bradley Sturt. Dynamic optimization with side information. *European Journal of Operational Research*, 2022.

Rafael Blanquero, Emilio Carrizosa, and Nuria Gómez-Vargas. Contextual uncertainty sets in robust linear optimization. 2023.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. Bail: Best-action imitation learning for batch deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:18353–18363, 2020.

Abhilash Reddy Chenreddy, Nymisha Bandi, and Erick Delage. Data-driven conditional robust optimization. *Advances in Neural Information Processing Systems*, 35:9525–9537, 2022.

Petros Christodoulou. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*, 2019.

Adrián Esteban-Pérez and Juan M. Morales. Distributionally robust stochastic programs with side information based on trimmings. *Mathematical Programming*, 195(1):1069–1105, 2022.

Angelos Filos, Panagiotis Tigas, Rowan McAllister, Yarin Gal, and Sergey Levine. Epistemic value estimation for risk-averse offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8073–8081, 2022.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. In *Foundations and Trends in Machine Learning*, volume 8, pages 359–483. Now Publishers Inc., 2015.

Dibya Ghosh, Anurag Ajay, Pulkit Agrawal, and Sergey Levine. Offline rl policies should be trained to be adaptive. In *International Conference on Machine Learning*, pages 7513–7530. PMLR, 2022.

Marc Goerigk and Jannis Kurtz. Data-driven robust optimization using deep neural networks. *Computers & Operations Research*, 151:106087, 2023.

Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez, Konrad Zolna, Rishabh Agarwal, Josh S Merel, Daniel J Mankowitz, Cosmin Paduraru, et al. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:7248–7259, 2020.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Adam Jelley, Trevor McInroe, Sam Devlin, and Amos Storkey. Efficient offline reinforcement learning: The critic is critical. *arXiv preprint arXiv:2406.13376*, 2024.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. When should we prefer offline reinforcement learning over behavioral cloning? *arXiv preprint arXiv:2204.05618*, 2022.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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.

Christopher McCord. *Data-driven dynamic optimization with auxiliary covariates*. PhD thesis, Massachusetts Institute of Technology, 2019.

Viet Anh Nguyen, Fan Zhang, Jose Blanchet, Erick Delage, and Yinyu Ye. Robustifying conditional portfolio decisions via optimal transport, 2021.

Shunichi Ohmori. A predictive prescription using minimum volume k-nearest neighbor enclosing ellipsoid and robust optimization. *Mathematics*, 9(2):119, 2021.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic neural networks. *Advances in Neural Information Processing Systems*, 36:2795–2823, 2023.

Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. A risk-sensitive perspective on model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 12345–12356, 2022.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Kajetan Schweighofer, Marius-constantin Dinu, Andreas Radler, Markus Hofmarcher, Vihang Prakash Patil, Angela Bitto-Nemling, Hamid Eghbal-zadeh, and Sepp Hochreiter. A dataset perspective on offline reinforcement learning. In *Conference on Lifelong Learning Agents*, pages 470–517. PMLR, 2022.

Laixi Shi and Yuejie Chi. Distributionally robust model-based offline reinforcement learning with near-optimal sample complexity. *Journal of Machine Learning Research*, 25(1):1–46, 2022.

Chunlin Sun, Linyu Liu, and Xiaocheng Li. Predict-then-calibrate: A new perspective of robust contextual lp. *Advances in Neural Information Processing Systems*, 36:17713–17741, 2023.

Irina Wang, Cole Becker, Bart Van Parys, and Bartolomeo Stellato. Learning for robust optimization. *arXiv preprint arXiv:2305.19225*, 2023.

Kai Wang and Alex Jacquillat. From classification to optimization: A scenario-based robust optimization approach. Available at SSRN 3734002, 2020.

Yeming Wen, Dustin Tran Han, and Jimmy Ba. Batchensemble: An alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

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

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:10299–10312, 2021.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. Minari, September 2024. URL https://doi.org/10.5281/zenodo.13767625.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

APPENDIX

This appendix provides literature context, theoretical foundations, algorithmic details, and extended empirical results that support our main findings.

We begin in Section with a review of related work on epistemic uncertainty modeling and robust offline reinforcement learning. Section analyzes the state visitation frequencies in the Machine Replacement problem under various behavior policies introduced in the main text. We further build on this example to study the sensitivity of the worst-case Q-function to the policy π_ϕ.

Section presents a formal lemma and proof showing that SAC-N is a special case of our proposed framework. Section derives closed-form expressions for the worst-case Q-vectors induced by convex hull and ellipsoidal sets.

Section provides pseudocode for the ERSAC algorithmic variants proposed in this work. Section describes the offline data generation process under different behavior policies. Section details the experimental setup, including training procedures and hyperparameters. Finally, Section presents full empirical results across environments, dataset sizes, and risk sensitivity levels, complementing the main text with additional tables and figures.

LITERATURE REVIEW

While the **motivation for offline RL** originates primarily from safety, cost, and deployment constraints in domains such as healthcare, robotics, and industrial control, recent work highlights its broader benefits, including improved generalization and sample efficiency when combined with online learning (Ball et al. (2023); Jelley et al. (2024)). Offline data can stabilize learning and accelerate convergence through pretraining or regularization (Kumar et al. (2022)). However, the absence of environment interaction exacerbates challenges like overestimation and error compounding, especially when using deep value function approximators. These failures are often attributed to epistemic uncertainty in out of distribution state-action pairs, where neural networks are known to make overconfident predictions (Lakshminarayanan et al. (2017); Kendall and Gal (2017)). Ensemble-based and Bayesian methods partially mitigate this by explicitly modeling uncertainty, highlighting the need for structured epistemic reasoning in offline settings.

Model-free methods primarily focus on constraining the learned policy or value estimates to remain within the support of the dataset, thereby mitigating extrapolation errors. One class of such methods, known as policy constraint methods, restricts the learned policy to stay close to the behavior policy. This reduces the likelihood of selecting actions not well represented in the data. Approaches like BCQ (Fujimoto et al. (2018)), BEAR (Kumar et al. (2019)), and BRAC (Wu et al. (2019)) explicitly enforce such constraints using divergence penalties or support matching. Another class focuses on value regularization, where conservative value estimates discourage overoptimistic Q-values for out-of-distribution actions. Notably, CQL (Kumar et al. (2020)) enforces a soft lower-bound on Q-values, while EDAC (An et al. (2021)) and other ensemble-based methods use Q-function diversity to reduce overestimation risk.

Model-based methods instead aim to learn an explicit model of the environment's dynamics, which can be used for policy learning or evaluation via simulated rollouts. Examples include MOPO (Yu et al. (2020)), which penalizes uncertainty in model rollouts, and MOReL (Kidambi et al. (2020)), which builds a pessimistic MDP based on model confidence. COMBO (Yu et al. (2021)) combines model-based rollouts with conservative value estimation to balance optimism and safety.

Other notable directions include trajectory optimization and decision-based methods, such as Decision Transformer (DT) (Chen et al. (2021)) and Implicit Q-Learning (IQL) (Kostrikov et al. (2021)), which cast offline RL as a supervised learning problem over sequences or value distributions. Additionally, imitation-based methods like BAIL (Chen et al. (2020)) interpolate between behavior cloning and value-based methods using uncertainty-aware selection of demonstration trajectories. We refer the reader to Levine et al. (2020); Prudencio et al. (2023) for comprehensive review of offline RL algorithms.

While **uncertainty quantification** is well studied in supervised learning and Bayesian RL (Ghavamzadeh et al. (2015)), its structured application in offline reinforcement learning remains underexplored. Traditional methods often conflate epistemic and aleatoric uncertainty or rely on coarse approximations such as ensemble minima, which can misrepresent uncertainty in regions with limited data. Recent work has begun to address these limitations by introducing methods that model epistemic uncertainty more explicitly. For example, Filos et al. (2022) propose Epistemic Value Estimation (EVE), which provides a task-aware mechanism for quantifying value uncertainty in offline settings. Similarly, Shi and Chi (2022) explore distributionally robust model-based offline RL using uncertainty sets over dynamics to improve robustness to model misspecification. Other approaches such as Panaganti et al. (2022) adopt a risk-sensitive view, incorporating epistemic uncertainty directly into policy optimization to avoid unsafe actions. Ensemble-based methods are a practical way to capture epistemic uncertainty. They have been used in both model-based settings (e.g., MOReL

Kidambi et al. (2020)) and model-free methods (e.g., EDAC An et al. (2021)) to stabilize learning by regularizing the Bellman backups or penalizing high-variance predictions. However, ensembles can be computationally expensive and coarse. More structured representations of epistemic uncertainty have been proposed using Epistemic Neural Networks (ENNs) (Osband et al. (2023)), which offer a flexible way to encode and sample from belief distributions over value functions. Building on these insights, our work introduces a structured, epistemic-robust alternative to ensemble pessimism by defining uncertainty sets over Q-values, allowing richer representations and more targeted conservatism in offline RL.

Additionally, **benchmarking offline RL** remains challenging due to limited dataset diversity. While D4RL (Fu et al. (2020)) and RL Unplugged (Gulcehre et al. (2020)) have improved standardization, existing benchmarks largely omit risk sensitive evaluation settings. Such behavior policies tend to handle high cost differently depending on whether they are risk averse or risk seeking. This implicit preference skews the data distribution and contributes to epistemic uncertainty, particularly in cases with less data. Despite its significance, there is currently no benchmark that allows systematic control over the risk sensitivity of the behavior policy to study its impact on offline RL performance. As a first step toward addressing this gap, we introduce a framework that enables controlled variation of behavioral risk preferences using dynamic expectiles. This allows us to generate offline datasets with adjustable risk profiles, facilitating principled evaluation of offline RL algorithms under different uncertainty conditions. Our proposed framework is aligned with recent efforts like the Minari platform proposed by Younis et al. (2024), but uniquely focuses on how risk sensitivity shapes epistemic uncertainty in offline datasets.

Building on these insights, this work introduces **Epistemic Robust Soft Actor-Critic (ERSAC)**, a unified framework for offline RL that models epistemic uncertainty through structured uncertainty sets over Q-values. By replacing ensemble based pessimism with compact and expressive set constructions such as box, convex hull, and ellipsoids, ERSAC enables conservative yet flexible value estimation and policy optimization. We show that SAC-N arises as a special case under box sets, and further extend the framework using **Epistemic Neural Networks (Epinet)** to construct ellipsoidal uncertainty sets in closed form, reducing runtime without sacrificing performance.

These contributions open several promising directions for future work, including integrating distributional robustness into set construction, incorporating risk-aware objectives, extending epistemic reasoning to multi-agent and hierarchical settings, and establishing theoretical guarantees such as generalization bounds and regret under epistemic uncertainty. Together, our results highlight the potential of structured and efficient epistemic modeling as a foundation for safe, generalizable, and scalable offline reinforcement learning.

MACHINE REPLACEMENT EXAMPLE

τ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0.1	0	0	0	0	0	0	0	0	1	1
0.5	0	0	0	0	0	0	1	1	1	1
0.9	0	0	0	0	0	1	1	1	1	1

Table 3: Optimal actions for each state under different expectile levels τ. Action 0 corresponds to progressing forward; Action 1 corresponds to jumping to state 1 with -100 reward.

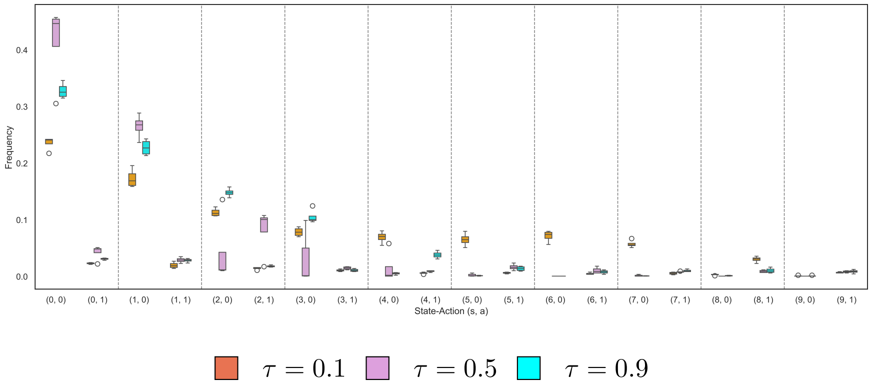
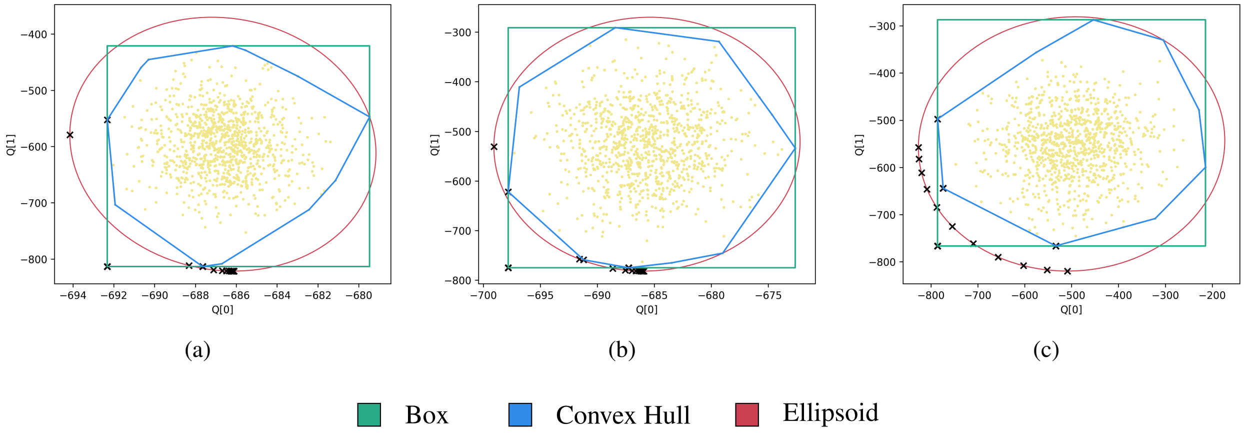


Figure 1: State visitation frequency distributions under different expectile policies.

Sensitivity of worst-case Q vector to π_ϕ

While the box set yields a fixed $q^*(s, \cdot\,; \phi)$ independent of the policy, both the convex hull and ellipsoidal sets adapt their minimizer $q^*(s, \cdot\,; \phi)$ to $\pi_\phi(\cdot \mid s)$. This flexibility introduces a richer learning dynamic, allowing the Bellman backup to respond differently depending on the current policy. This behavior can be viewed from a game-theoretic point of view. At each state s, the agent proposes a policy $\pi_\phi(\cdot \mid s)$, and an adversary selects the worst-case Q-vector $q^*(s, \cdot\,; \phi) \in \mathcal{U}_\theta(s)$ that minimizes the expected return $\langle \pi_\phi(\cdot \mid s), q \rangle$. When the uncertainty set contains multiple non-dominated extremal points, as is the case for convex hulls and ellipsoids, the Bellman update becomes more responsive capable of adjusting its conservativeness based on the agent's action preferences. To illustrate this, consider the Machine Replacement example discussed above. Figure 2 highlights this adaptivity across selected states by comparing the q^* responses of the three sets $\mathcal{U}_{\text{box}}(s), \mathcal{U}_{\text{hull}}(s)$ and $\mathcal{U}_{\text{ell}}(s)$ as the policy π varies uniformly over the probability simplex. This behavior leads to a more expressive training process that is sensitive to the epistemic structure captured by the generative model.



(a)

(b)

(c)

Figure 2: (a)–(c): Uncertainty sets and worst-case policy evaluations for states 0, 5, and 10 in the machine replacement example at epoch 1. Each subplot illustrates the distribution of ensemble Q-values along with the corresponding box, convex hull, and ellipsoidal uncertainty sets. Markers X indicate the worst-case Q-value q^* under different policies π.

This adaptivity is particularly important in offline settings, where data coverage is often limited or biased. Structured uncertainty sets enable value estimates that are conservative in underexplored regions while remaining responsive in well-covered ones, leading to improved generalization without excessive pessimism.

The construction of these sets connects with the recent evolving literature in Estimate-then-Optimize Conditional Robust Optimization (CRO). One line of work as proposed in Chenreddy et al. (2022); Goerigk and Kurtz (2023); Ohmori (2021); Sun et al. (2023); Blanquero et al. (2023) focuses on calibrating uncertainty sets over realizations drawn from a conditional distribution $F(q \mid s)$. These methods construct high-probability sets $\mathcal{U}(s) \subset \mathbb{R}^d$ such that for a random realization $q \sim F(\cdot \mid s)$, it holds that $\mathbb{P}(q \in \mathcal{U}(s)) \geq 1 - \delta$. Such calibrated sets enable robust decisions of the form $\max_{\pi \in \Pi} \min_{q \in \mathcal{U}(s)} \pi^\top q$, that ensure performance against probable realizations of the uncertain quantity q, conditioned on covariates s.

A second line of work, common in distributionally robust optimization and robust RL constructs ambiguity sets over the distribution $F(\cdot \mid s)$ itself, e.g., using moment constraints, Wasserstein balls, or scenario-based support ((Bertsimas et al., 2022; McCord, 2019; Wang and Jacquillat, 2020; Wang et al., 2023; Nguyen et al., 2021; Esteban-Pérez and Morales, 2022)). In this setting, one solves:

$$\max_{\pi \in \Pi} \min_{F \in \mathcal{F}(s)} \mathbb{E}_{q \sim F}[\pi^\top q] = \max_{\pi \in \Pi} \min_{\bar{q} \in \mathcal{U}(s)} \pi^\top \bar{q},$$

where $\mathcal{F}(s)$ is an ambiguity set over distributions and $\mathcal{U}(s) := \{\mathbb{E}_{q \sim F}[q] : F \in \mathcal{F}(s)\}$ is the implied uncertainty set over expected values.

Our work aligns more closely with the former, wherein we directly parameterize and sample from a learned conditional distribution $\widehat{F}^q_\theta(s)$, and define a structured uncertainty set $\mathcal{U}(\widehat{F}^q_\theta(s))$ over sampled realizations $q \sim \widehat{F}^q_\theta(s)$. This allows us to reason about epistemic variability in Q-values without requiring a full ambiguity set over $F^q_\theta(s)$. Bridging these two lines of work could lead to rich formulations for epistemically robust reinforcement learning, which we leave for future work.

Proof for Proposition1

We begin by analyzing the robust estimator term present in both the conservative target value (8) and the policy loss (10): $\min_{q \in \mathcal{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q \rangle$. Given that the uncertainty set is defined as a coordinate-wise product box and that $\pi_\phi(\cdot \mid s) \geq 0$, the minimum must be achieved at the coordinate-wise lower bound:

$$\begin{aligned} q^*(a) &= \operatorname{essinf}_{\tilde{q} \sim F^q_\theta(s)} [\tilde{q}(a)] \\ &= \operatorname{essinf}_{\tilde{i} \sim U(N)} [Q_{\theta_{\tilde{i}}}(s, a)] \\ &= \min_{i \in [N]} Q_{\theta_i}(s, a), \quad \forall a \in \mathcal{A}. \end{aligned}$$

The robust evaluation then becomes,

$$\begin{aligned} \min_{q \in \mathcal{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q \rangle &= \sum_{a \in \mathcal{A}} \pi_\phi(a \mid s) \min_{i \in [N]} Q_{\theta_i}(s, a) \\ &= \mathbb{E}_{a \sim \pi_\phi(\cdot|s)} \left[\min_{i \in [N]} Q_{\theta_i}(s, a) \right] \end{aligned}$$

Hence, the conservative target value becomes

$$\begin{aligned} y(r, s') &= r + \gamma \, \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} \left[\min_{i \in [N]} Q_{\theta_i}(s', a') - \alpha \log \pi_\phi(a' \mid s') \right] \\ &= \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} \left[r + \gamma \big(\min_{i \in [N]} Q_{\theta_i}(s', a') - \alpha \log \pi_\phi(a' \mid s') \big) \right] \\ &= \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} [y(r, s', a')] \end{aligned}$$

We thus have that,

$$\begin{aligned} \mathcal{L}^R_Q(\theta) &= \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ \tilde{q} \sim F^q_\theta(s)}} \left[\big(\tilde{q}(a) - y(r, s') \big)^2 \right] \\ &= \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ \tilde{q} \sim F^q_\theta(s)}} \left[\tilde{q}(a)^2 - 2\tilde{q}(a) \, \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} [y(r, s', a')] + \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} [y(r, s', a')]^2 \right] \\ &= \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ \tilde{q} \sim F^q_\theta(s)}} \left[\tilde{q}(a)^2 - 2\tilde{q}(a) \, \mathbb{E}_{a'} [y(r, s', a')] + \mathbb{E}_{a'} \left[y(r, s', a')^2 \right] \right] \\ &\quad + \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\mathbb{E}_{a'} [y(r, s', a')]^2 - \mathbb{E}_{a'} [y(r, s', a')^2] \right] \\ &= \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ \tilde{q} \sim F^q_\theta(s) \\ a' \sim \pi_\phi(\cdot|s')}} \left[\tilde{q}(a)^2 - 2\tilde{q}(a) \, y(r, s', a') + y(r, s', a')^2 \right] + C \\ &= \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ \tilde{q} \sim F^q_\theta(s) \\ a' \sim \pi_\phi(\cdot|s')}} \left[\big(\tilde{q}(a) - y(r, s', a') \big)^2 \right] + C \\ &= \frac{1}{N} \sum_i \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ a' \sim \pi_\phi(\cdot|s')}} \left[\big(Q_{\theta_i}(s, a) - y(r, s', a') \big)^2 \right] + C \\ &= \frac{1}{N} \sum_i \mathcal{L}_Q(\theta_i) + C \end{aligned}$$

where

$$C := \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\big(\mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} [y(r, s', a')] \big)^2 \right] - \mathbb{E}_{\substack{(s,a,r,s') \sim \mathcal{D} \\ a' \sim \pi_\phi(\cdot|s')}} \left[y(r, s', a')^2 \right]$$

due to $\tilde{q}(a)$ being independent of $y(r, s', a')$ given (s, a, r, s').

On the other hand, we have that:

$$\begin{aligned}
\mathcal{J}_\pi^R(\phi) &= \mathbb{E}_{s\sim\mathcal{D},\ a\sim\pi_\phi(\cdot|s)}\left[\min_{q\in\mathcal{U}_\theta(s)}\langle\pi_\phi(\cdot\mid s), q\rangle - \alpha\log\pi_\phi(a\mid s)\right] \\
&= \mathbb{E}_{s\sim\mathcal{D},\ a\sim\pi_\phi(\cdot|s)}\left[\mathbb{E}_{a'\sim\pi_\phi(\cdot|s)}\big[\min_{i\in[N]} Q_{\theta_i}(s, a')\big] - \alpha\log\pi_\phi(a\mid s)\right] \\
&= \mathbb{E}_{s\sim\mathcal{D},\ a\sim\pi_\phi(\cdot|s)}\left[\min_{i\in[N]} Q_{\theta_i}(s, a) - \alpha\log\pi_\phi(a\mid s)\right] \\
&= \mathcal{J}_\pi(\phi).
\end{aligned}$$

This completes our proof.

Derivations of Worst-Case Q-vector Expressions

This section provides derivations supporting the closed-form expressions of the worst-case Q-vector $q^*(s, \cdot; \phi)$ under the convex hull and ellipsoidal uncertainty sets, as referenced in Section . These derivations clarify how the worst-case backup depends on the policy π_ϕ.

Convex Hull Set

The worst-case expected Q-value over the convex hull uncertainty set is given by:

$$\begin{aligned}
&\min_{q\in\mathcal{U}_{\text{hull}}(\widehat{F}_\theta^q(s))} \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[q(a)] \\
&\qquad = \min_{\substack{\lambda\geq 0 \\ \sum_{i=1}^N\lambda_i=1}} \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\sum_{i=1}^N \lambda_i\,\mathfrak{q}_\theta(s, a, \tilde{z}_i)\right] \\
&\qquad = \min_{\substack{\lambda\geq 0 \\ \sum_{i=1}^N\lambda_i=1}} \sum_{i=1}^N \lambda_i\,\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\mathfrak{q}_\theta(s, a, \tilde{z}_i)\right] \\
&\qquad \geq \min_{i\in[N]} \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\mathfrak{q}_\theta(s, a, \tilde{z}_i)\right] \\
&\qquad = \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\mathfrak{q}_\theta(s, a, z^*(s, \phi))\right],
\end{aligned}$$

where $z^*(s, \phi) \in \arg\min_i \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\mathfrak{q}_\theta(s, a, \tilde{z}_i)\right]$.

Ellipsoidal Set

For the ellipsoidal set, we consider the constrained optimization problem:

$$\begin{aligned}
&\min_{q\in\mathcal{U}_{\text{ell}}(\widehat{F}_\theta^q(s))} \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[q(a)] \\
&= \min_{\substack{q: \\ (q-\hat{\mu}(s))^\top\widehat{\Sigma}(s)^{-1}(q-\hat{\mu}(s))\leq\widehat{\Upsilon}(s)^2}} \langle\pi_\phi(\cdot\mid s), q\rangle \\
&= \min_{\substack{\zeta: \\ \|\zeta\|\leq\widehat{\Upsilon}(s)}} \langle\pi_\phi(\cdot\mid s), \hat{\mu}(s) + \widehat{\Sigma}^{1/2}(s)\,\zeta\rangle \\
&\geq \langle\pi_\phi(\cdot\mid s), \hat{\mu}(s)\rangle - \widehat{\Upsilon}(s)\left\|\widehat{\Sigma}^{1/2}(s)\,\pi_\phi(\cdot\mid s)\right\| \\
&= \left\langle \pi_\phi(\cdot\mid s),\ \hat{\mu}(s) - \widehat{\Upsilon}(s)\cdot\frac{\widehat{\Sigma}(s)\,\pi_\phi(\cdot\mid s)}{\left\|\widehat{\Sigma}^{1/2}(s)\,\pi_\phi(\cdot\mid s)\right\|}\right\rangle,
\end{aligned}$$

where we applied the Cauchy-Schwarz inequality in the third step. This expression matches the closed-form solution for the worst-case Q-vector under the ellipsoidal uncertainty set.

Algorithmic Implementation Details

In this section, we present the pseudocode for the algorithms discussed in the main paper.

Algorithm 1: Epistemic Robust SAC Training

```
Input: Initial policy parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$, offline data replay buffer $\mathcal{D}$, learning rates $\eta_Q, \eta_\pi$, target update rate $\tau$
for each epoch do
    Sample minibatch $\mathcal{B} := \{(s, a, r, s')\}$ from $\mathcal{D}$
    Compute target:
        $$y(r, s') \leftarrow r + \gamma\left(\min_{q \in \mathcal{U}_{\theta'}(s')} \langle \pi_\phi(\cdot \mid s'),\, q\rangle - \alpha\, \mathbb{E}_{a' \sim \pi_\phi}[\log \pi_\phi(a' \mid s')]\right)$$
    Critic update:
        $$\theta \leftarrow \theta - \eta_Q \cdot \tfrac{2}{|\mathcal{B}|} \sum_{(s,a,r,s') \in \mathcal{B}} \mathbb{E}_{\tilde{z} \sim F_z}\left[(\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s')) \cdot \nabla_\theta \mathfrak{q}_\theta(s, a, \tilde{z})\right]$$
    Compute worst-case $q^*$ vectors:
        $$q^*(s, \cdot\,; \phi) \leftarrow \arg \min_{q \in \mathcal{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q\rangle$$
    Actor update:
        $$\phi \leftarrow \phi + \eta_\pi \cdot \tfrac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \left( \sum_{a \in \mathcal{A}} q^*(s, a\,; \phi)\, \nabla_\phi \pi_\phi(a \mid s) - \alpha\, \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(\cdot \mid s)}[\log \pi_\phi(a \mid s)]\right)$$
    Update target network: $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$
```

Algorithm 2: Sample-based Epistemic Robust SAC with Box (ERSAC-B) and Convex Hull (ERSAC-CH) Sets

```
Input: Initial policy parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$, offline data buffer $\mathcal{D}$, learning rates $\eta_Q, \eta_\pi$,
       target update rate $\tau$, sample size $N$
for each epoch do
    Sample minibatch $\mathcal{B} := \{(s, a, r, s')\}$ from $\mathcal{D}$
    Sample $N$ i.i.d. latent variables $\{\tilde{z}_i\}_{i=1}^{N}$ from $F_z$

    Construct sampled Q-values:

    $\mathcal{Q}(s) \leftarrow \{\mathfrak{q}_\theta(s, \cdot, \tilde{z}_i)\}_{i=1}^{N}$

    $\mathcal{Q}(s') \leftarrow \{\mathfrak{q}_{\theta'}(s', \cdot, \tilde{z}_i)\}_{i=1}^{N}$

    Construct robust target:

    // Box set
        $y_{\text{box}}(r, s') = r + \gamma\left(\sum_{a \in \mathcal{A}} \pi_\phi(a \mid s') \cdot \min_{i \in [N]} \mathfrak{q}_{\theta'}(s', a, \tilde{z}_i) - \alpha \sum_{a \in \mathcal{A}} \pi_\phi(a \mid s') \log \pi_\phi(a \mid s')\right)$
    // Convex Hull set
        $y_{\text{hull}}(r, s') = r + \gamma\left(\min_{i \in [N]} \sum_{a \in \mathcal{A}} \pi_\phi(a \mid s') \cdot \mathfrak{q}_{\theta'}(s', a, \tilde{z}_i) - \alpha \sum_{a \in \mathcal{A}} \pi_\phi(a \mid s') \log \pi_\phi(a \mid s')\right)$
    Critic update (common):
        $\theta \leftarrow \theta - \eta_Q \cdot \frac{2}{|\mathcal{B}|} \sum_{(s,a,r,s') \in \mathcal{B}} \mathbb{E}_{\tilde{z} \sim F_z}\big[\,(\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s')) \cdot \nabla_\theta \mathfrak{q}_\theta(s, a, \tilde{z})\big]$
    Actor update:

    // Box set
        $\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \left(\sum_{a \in \mathcal{A}} \min_{i \in [N]} \mathfrak{q}_\theta(s, a, \tilde{z}_i) \nabla_\phi \pi_\phi(a \mid s) - \alpha\, \nabla_\phi\, \mathbb{E}_{a \sim \pi_\phi(\cdot|s)}\left[\log \pi_\phi(a \mid s)\right]\right)$
    // Convex Hull set
        $i^* = \arg\min_{i \in [N]} \sum_{a \in \mathcal{A}} \pi_\phi(a \mid s) \cdot \mathfrak{q}_\theta(s, a, \tilde{z}_i)$
        $\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \sum_{a \in \mathcal{A}} \mathfrak{q}_\theta(s, a, \tilde{z}_{i^*}) \cdot \nabla_\phi \pi_\phi(a \mid s) - \alpha \cdot \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(\cdot|s)}\left[\log \pi_\phi(a \mid s)\right]$
    Target network update: $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$
```

Algorithm 3: Sample-based Epistemic Robust SAC with Ellipsoidal Uncertainty (ERSAC-E)

```
Input: Initial policy parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$, offline data replay buffer $\mathcal{D}$, learning rates
       $\eta_Q, \eta_\pi$, target update rate $\tau$, sample size $N$
for each epoch do
    Sample minibatch $\mathcal{B} := \{(s, a, r, s')\}$ from $\mathcal{D}$
     Sample $N$ i.i.d. realizations $\{\tilde{z}_i\}$ from $F_z$
     $\hat{\mu}(s) \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathfrak{q}_\theta(s, \cdot, \tilde{z}_i)$
     $\widehat{\Sigma}(s) \leftarrow \frac{1}{N} \sum_{i=1}^{N} (\mathfrak{q}_\theta(s, \cdot, \tilde{z}_i) - \hat{\mu}(s))(\mathfrak{q}_\theta(s, \cdot, \tilde{z}_i) - \hat{\mu}(s))^\top$
     $\widehat{\Upsilon}(s) \leftarrow \inf\{\Upsilon | \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{(\mathfrak{q}_\theta(s, \cdot, \tilde{z}_i) - \hat{\mu}(s))^\top \widehat{\Sigma}(s)^{-1} (\mathfrak{q}_\theta(s, \cdot, \tilde{z}_i) - \hat{\mu}(s)) \leq \Upsilon^2\} \geq \upsilon\}$
     $\hat{\mu}(s') \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathfrak{q}_{\theta'}(s', \cdot, \tilde{z}_i)$
     $\widehat{\Sigma}(s') \leftarrow \frac{1}{N} \sum_{i=1}^{N} (\mathfrak{q}_{\theta'}(s', \cdot, \tilde{z}_i) - \hat{\mu}(s'))(\mathfrak{q}_{\theta'}(s', \cdot, \tilde{z}_i) - \hat{\mu}(s'))^\top$
     $\widehat{\Upsilon}(s') \leftarrow \inf\{\Upsilon | \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{(\mathfrak{q}_{\theta'}(s', \cdot, \tilde{z}_i) - \hat{\mu}(s'))^\top \widehat{\Sigma}(s')^{-1} (\mathfrak{q}_{\theta'}(s', \cdot, \tilde{z}_i) - \hat{\mu}(s')) \leq \Upsilon^2\} \geq \upsilon\}$
     Compute target:
         $y(r, s') \leftarrow r + \gamma \Big( \langle \pi_\phi(\cdot \mid s'),\, \hat{\mu}(s') \rangle - \widehat{\Upsilon}(s') \left\| \widehat{\Sigma}^{1/2}(s')\, \pi_\phi(\cdot \mid s') \right\| - \alpha\, \mathbb{E}_{a' \sim \pi_\phi} \left[ \log \pi_\phi(a' \mid s') \right] \Big)$
    Critic update:
         $\theta \leftarrow \theta - \eta_Q \cdot \frac{2}{|\mathcal{B}|} \sum_{(s,a,r,s') \in \mathcal{B}} \mathbb{E}_{\tilde{z} \sim F_z} \left[ (\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s')) \cdot \nabla_\theta \mathfrak{q}_\theta(s, a, \tilde{z}) \right]$
    Actor update:
         $\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \sum_{a \in \mathcal{A}} \left( \hat{\mu}(s, a) - \widehat{\Upsilon}(s) \cdot \frac{[\widehat{\Sigma}(s) \pi_\phi(\cdot|s)](a)}{\|\widehat{\Sigma}^{1/2}(s) \pi_\phi(\cdot|s)\|} \right) \nabla_\phi \pi_\phi(a \mid s) - \alpha\, \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(\cdot|s)} \left[ \log \pi_\phi(a \mid s) \right]$
    Update target network: $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$
```

Algorithm 4: Sample-based ERSAC with Ellipsoidal Uncertainty using Epinet (ERSAC-E-Epi)

```
Input: Initial policy parameters $\phi$; Q-network parameters $\theta = (\theta_\mu, \theta_\sigma)$; target network parameters $\theta' = (\theta'_\mu, \theta'_\sigma)$; offline data buffer $\mathcal{D}$; learning rates $\eta_Q, \eta_\pi$; target update rate $\tau$; noise scale $\bar{\sigma}$; regularization coefficients $\lambda_\mu, \lambda_\sigma$; sample size $N$
for each epoch do
    Sample minibatch $\bar{\mathcal{B}} := \{(s, a, r, s', c)\}$ from augmented buffer $\bar{\mathcal{D}}$ where $c \sim \mathrm{Unif}(\mathbb{S}^{d_z})$ Sample $N$ i.i.d. latent indices $\{\tilde{z}_i\}_{i=1}^N \sim \mathcal{N}(0, I)$
    Construct uncertainty set (Epinet-based ellipsoid):
        $\hat{\mu}(s') \leftarrow \mu_{\theta'_\mu}(s')$
        $\bar{\sigma}_{\theta'}(s', a) \leftarrow \bar{\sigma}^L_{\theta'_\sigma}(\psi_{\theta'_\mu}(s'), a) + \bar{\sigma}^P(\psi_{\theta'_\mu}(s'), a)$
        $\Sigma_{\theta'}(s')_{a,a'} \leftarrow \langle \bar{\sigma}_{\theta'}(s', a), \bar{\sigma}_{\theta'}(s', a') \rangle$
    Compute robust target:
        $$y(r, s') \leftarrow r + \gamma \left( \langle \pi_\phi(\cdot \mid s'), \hat{\mu}(s') \rangle - \rho \left\| \Sigma^{1/2}_{\theta'}(s') \pi_\phi(\cdot \mid s') \right\|_2 - \alpha \, \mathbb{E}_{a' \sim \pi_\phi} \left[ \log \pi_\phi(a' \mid s') \right] \right)$$
    Critic update:
        $$\theta_\mu \leftarrow \theta_\mu - 2\eta_Q \cdot \frac{1}{|\bar{\mathcal{B}}|} \sum_{(s,a,r,s',c) \in \bar{\mathcal{B}}} \mathbb{E}_{\tilde{z} \sim \mathcal{N}(0,I)} \left[ (\mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s') - \bar{\sigma} \langle c, \tilde{z} \rangle) \cdot \nabla_{\theta_\mu} \mu_{\theta_\mu}(s, a) \right] + 2\lambda_\mu \theta_\mu$$
        $$\theta_\sigma \leftarrow \theta_\sigma - 2\eta_Q \cdot \frac{1}{|\bar{\mathcal{B}}|} \sum_{(s,a,r,s',c) \in \bar{\mathcal{B}}} \mathbb{E}_{\tilde{z} \sim \mathcal{N}(0,I)} \left[ \left( \mathfrak{q}_\theta(s, a, \tilde{z}) - y(r, s') - \bar{\sigma} \langle c, \tilde{z} \rangle \right) \cdot \nabla_{\theta_\sigma} \sigma^L_{\theta_\sigma}(\psi_{\theta_\mu}(s), a, \tilde{z}) \right] + 2\lambda_\sigma \theta_\sigma$$
    Actor update:
        $$\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\bar{\mathcal{B}}|} \sum_{s \in \bar{\mathcal{B}}} \left[ \sum_{a \in \mathcal{A}} \left( \hat{\mu}(s, a) - \rho \cdot \frac{\Sigma_\theta(s) \pi_\phi(a \mid s)}{\left\| \Sigma^{1/2}_\theta(s) \pi_\phi(\cdot \mid s) \right\|} \right) \nabla_\phi \pi_\phi(a \mid s) - \alpha \cdot \nabla_\phi \mathbb{E}_{a \sim \pi_\phi} [\log \pi_\phi(a \mid s)] \right]$$
    Update target networks: $\theta' \leftarrow \tau \cdot \theta + (1 - \tau) \cdot \theta'$
```

Risk-Sensitive Offline Data Generation

Algorithm 5: Offline Data Generation via Dynamic Expectile Risk Policies

```
Input: Environment $\mathcal{M}$; risk level $\tau \in (0,1)$; dataset size $N_\mathcal{D}$; initial policy parameters $\phi$, Q parameters $\theta$, target Q
        parameters $\theta'$, learning rates $\eta_Q$, $\eta_\pi$; exploration rate $\epsilon$; number of sample $N_s$ for $P(\cdot|s,a)$ approximation
Output: Offline dataset $\mathcal{D}$
Initialize policy parameters $\phi$ and value function parameters $\theta$
for each epoch do
    Initialize state $s$
    while Episode not done do
        Sample transition $(s,a,r,s')$ by executing current policy $\pi_\phi$ in environment $\mathcal{M}$
        Compute expectile target:
            $y \leftarrow \sup\left\{z :\ \mathbb{E}_{s'\sim\hat{p}_{N_s}(\cdot|s,a)}\left[\left|\tau - \mathbb{I}\left(z < r + \gamma \max_{a'} Q_{\theta'}(s',a')\right)\right| \cdot \left(z - r - \gamma \max_{a'} Q_{\theta'}(s',a')\right)\right] \le 0\right\}$
        where $\hat{p}_{N_s}(\cdot|s,a)$ is empirical distribution of $N_s$ resampling of the transition from $(s,a)$
        Update value function:
            $\theta \leftarrow \theta - \eta_Q \cdot \nabla_\theta \left(Q_\theta(s,a) - y\right)^2$
        Update policy:
            $\phi \leftarrow \phi + \eta_\pi \cdot \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\nabla_\phi \log \pi_\phi(a|s) \cdot Q_\theta(s,a)\right]$
        Move to next state: $s \leftarrow s'$
    Update target network: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
Offline Data Collection with $\epsilon$-Greedy Exploration:
Initialize empty dataset $\mathcal{D} \leftarrow \emptyset$
while $|\mathcal{D}| < N_\mathcal{D}$ do
    Observe state $s$ from environment $\mathcal{M}$
    if RandomUniform(0,1) < $\epsilon$ then
        Sample action $a \sim$ Uniform($\mathcal{A}$)
    else
        Sample action $a \sim \pi_\phi(\cdot|s)$
    Execute action $a$ in environment to observe $r$ and $s'$
    Store $(s,a,r,s')$ in buffer $\mathcal{D}$
return Dataset $\mathcal{D}$
```

Training algorithm details

We evaluate all algorithms on a tabular Machine Replacement MDP with $S = 10$ states and $A = 2$ actions. Transition dynamics are defined probabilistically, with increasing expected costs for continued operation and a reset mechanism triggered by replacement actions. Rewards are state- and transition-dependent, with negative values to simulate maintenance costs and catastrophic penalties for failure.

To construct behavior policies, we implement risk-sensitive value iteration using the expectile risk measure at levels $\tau \in \{0.1, 0.5, 0.9\}$. Expectile backups are computed by solving a convex root-finding problem for each state-action pair. Policies are derived via one-hot argmax over the resulting Q-values.

We generate offline trajectories using the expectile-optimal policy π_τ for each τ. At each step, with probability 0.1, a uniformly random action is taken for exploration. We vary the number of transitions $M \in \{100, 1000, 10000\}$ and use ten random seeds per setting. Each trajectory entry records (s, a, s', r).

We evaluate three risk-sensitive SAC-N variants using $N = 100$ Q-ensemble members. Each method includes entropy regularization with coefficient $\alpha = 0.01$ and actor-critic learning rates $\eta_q = \eta_\pi = 0.01$. Target networks are updated using Polyak averaging with $\tau = 0.005$.

We report normalized returns with respect to the optimal and random policies:

$$\text{Normalized Return} = \frac{V_{\text{eval}} - V_{\text{random}}}{V_{\text{optimal}} - V_{\text{random}}},$$

Env	DS	τ	SAC-N	CH-N	Ell-N	Ell_0.9-N	Beh. Policy
Machine Replacement	10×	0.1	$\underline{80 \pm 3}$	85 ± 2	87 ± 1	$\mathbf{88 \pm 2}$	86 ± 3
	100×	0.1	97 ± 1	97 ± 1	$\underline{95 \pm 2}$	96 ± 2	86 ± 3
	1000×	0.1	98 ± 2	98 ± 2	96 ± 2	96 ± 1	86 ± 3
	10×	0.5	$\underline{87 \pm 2}$	88 ± 2	90 ± 2	$\mathbf{91 \pm 2}$	100 ± 0
	100×	0.5	97 ± 1	$\mathbf{98 \pm 1}$	$\underline{92 \pm 2}$	94 ± 2	100 ± 0
	1000×	0.5	98 ± 2	98 ± 2	98 ± 2	$\mathbf{99 \pm 0}$	100 ± 0
	10×	0.9	$\underline{85 \pm 2}$	86 ± 2	90 ± 2	90 ± 2	92 ± 2
	100×	0.9	96 ± 2	96 ± 2	$\underline{95 \pm 2}$	96 ± 2	92 ± 2
	1000×	0.9	96 ± 2	96 ± 2	96 ± 2	96 ± 1	92 ± 2
RiverSwim	10×	0.1	$\underline{37 \pm 4}$	64 ± 2	57 ± 3	$\mathbf{66 \pm 3}$	-20 ± 3
	100×	0.1	$\underline{92 \pm 2}$	94 ± 2	94 ± 3	94 ± 3	-20 ± 3
	1000×	0.1	$\underline{99 \pm 1}$	100 ± 0	100 ± 0	100 ± 0	-20 ± 3
	10×	0.5	$\underline{56 \pm 2}$	60 ± 2	60 ± 2	$\mathbf{62 \pm 1}$	100 ± 0
	100×	0.5	$\underline{97 \pm 2}$	99 ± 1	98 ± 1	99 ± 1	100 ± 0
	1000×	0.5	99 ± 1	99 ± 1	100 ± 0	100 ± 0	100 ± 0
	10×	0.9	49 ± 2	49 ± 4	$\underline{48 \pm 1}$	$\mathbf{52 \pm 3}$	34 ± 4
	100×	0.9	99 ± 1	99 ± 1	$\mathbf{100 \pm 0}$	99 ± 1	34 ± 4
	1000×	0.9	99 ± 1	99 ± 1	100 ± 0	100 ± 0	34 ± 4

Table 4: Normalized returns with 90% confidence interval achieved by SAC-N, CH-N, Ell-N, and Ell_0.9-N across dataset sizes $\{10\times, 100\times, 1000\times\}$ and behavior policy risk levels $\tau \in \{0.1, 0.5, 0.9\}$ in the Machine Replacement and RiverSwim environments. Scores are computed over 10 evaluation seeds and normalized relative to the random and optimal policy baselines. Bold and underline highlight respectively the best and worst performing method when the margin is larger or equal to one. The final column reports the return of the behavior policy used to generate the offline data.

averaged over 1000 episodes. Returns are discounted with $\gamma = 0.9$. We repeat all experiments across ten seeds and report the mean and standard deviation. All code is implemented in Pytorch and NumPy using vectorized operations. Root-finding in expectile computation uses a bisection method with machine epsilon tolerance.

Detailed results

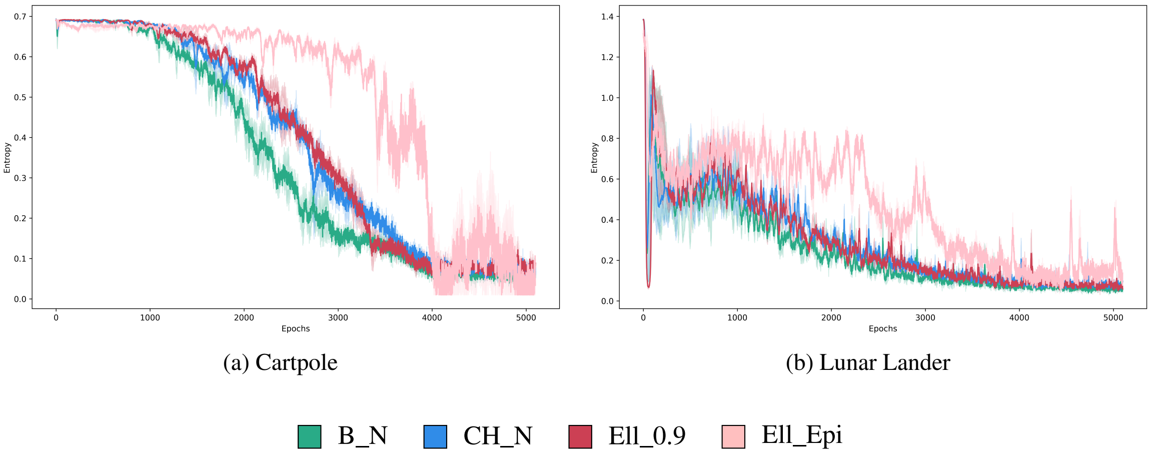
This section presents more details about the experiments that are discussed in the main text of the paper. Table 4 presents additional details on the experiments involving the tabular tasks (i.e., Machine Replacement and RiverSwim). Table 5 presents more detailed statistics about the experiments involving the CartPole and LunarLander Gym environments. Table 6 follows with a report of the runtimes (in s/epoch) of the five offline RL algorithms in the LunarLander Gym. Finally, Figure 3 compares the entropy of the policies obtained from four ER-SAC variants during each epoch of the training. As remarked in the main text, Box-based methods (B-N) maintain consistently lower entropy than **CH-N**, **Ell-N**, and **Ell-Epi**.

Env	DS	τ	SAC-N	CH-N	Ell_0.9-N	Ell-Epi	Ell-Epi*	Beh. Policy
CartPole	1k	0.1	84 ± 3	81 ± 2	$\mathbf{86} \pm \mathbf{1}$	84 ± 1	85 ± 2	86 ± 2
	10k	0.1	92 ± 2	94 ± 2	100 ± 0	100 ± 0	100 ± 0	86 ± 2
	100k	0.1	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	86 ± 2
	1k	0.5	70 ± 2	72 ± 1	$\mathbf{73} \pm \mathbf{3}$	72 ± 2	71 ± 2	100 ± 0
	10k	0.5	97 ± 2	99 ± 1	100 ± 0	100 ± 0	100 ± 0	100 ± 0
	100k	0.5	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0
	1k	0.9	73 ± 2	70 ± 3	78 ± 2	$\mathbf{80} \pm \mathbf{1}$	75 ± 2	83 ± 2
	10k	0.9	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	83 ± 2
	100k	0.9	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	83 ± 2
LunarLander	1k	0.1	72 ± 1	77 ± 1	98 ± 2	97 ± 3	98 ± 2	94 ± 3
	10k	0.1	94 ± 2	98 ± 1	102 ± 1	102 ± 3	$\mathbf{103} \pm \mathbf{1}$	94 ± 2
	100k	0.1	99 ± 1	100 ± 3	106 ± 1	$\mathbf{110} \pm \mathbf{3}$	108 ± 1	94 ± 2
	1k	0.5	68 ± 3	73 ± 3	96 ± 3	95 ± 1	$\mathbf{97} \pm \mathbf{1}$	100 ± 2
	10k	0.5	93 ± 3	99 ± 1	100 ± 1	99 ± 1	$\mathbf{102} \pm \mathbf{1}$	100 ± 2
	100k	0.5	98 ± 2	100 ± 1	102 ± 2	$\mathbf{108} \pm \mathbf{2}$	105 ± 2	100 ± 2
	1k	0.9	67 ± 2	73 ± 2	97 ± 2	$\mathbf{98} \pm \mathbf{2}$	97 ± 2	78 ± 3
	10k	0.9	92 ± 2	92 ± 3	101 ± 2	100 ± 4	$\mathbf{102} \pm \mathbf{2}$	78 ± 3
	100k	0.9	98 ± 2	101 ± 2	103 ± 1	104 ± 2	$\mathbf{105} \pm \mathbf{1}$	78 ± 3

Table 5: Normalized returns with 90% confidence intervals achieved by the five algorithms across dataset sizes $\{1\text{k}, 10\text{k}, 100\text{k}\}$ and behavior-policy risk levels $\tau \in \{0.1, 0.5, 0.9\}$ in CartPole and LunarLander. Scores are averaged over 10 evaluation seeds and normalized against random and optimal baselines. Bold and underline highlight respectively the best and worst performing method when the margin is larger or equal to one.

Model	SAC-N	CH-N	Ell_0.9-N	Ell-Epi	Ell-Epi*
Runtime (s/epoch)	0.35	0.42	0.56	0.60	0.10

Table 6: Runtime per training epoch for each model in LunarLander with 100,000 offline transitions and $\tau = 0.5$, averaged over 10 seeds.



(a) Cartpole

(b) Lunar Lander

Figure 3: Policy entropy during training across B_N, CH_N, Ell_0.9, and Ell_Epi models in the CartPole and LunarLander environments. Entropy is computed per epoch and averaged over 10 evaluation seeds. Lower entropy indicates more confident, deterministic policies, while higher entropy reflects greater stochasticity in policies.