Bootstrap Ensemble Uncertainty for State-Adaptive Regularization in Offline Reinforcement Learning

Rishav Rishav*
University of Calgary, Mila

Vincent Michalski†
University of Montreal, Mila

Samira Ebrahimi Kahou†
University of Calgary, Mila, Canada CIFAR AI Chair

Abstract

Offline reinforcement learning (RL) aims to learn an effective policy from a static dataset. An important challenge, however, is handling of out-of-distribution (OOD) actions that are poorly represented in the data. OOD actions lead to unreliable value estimates and subsequent policy degradation. Traditional approaches address this through regularization, but apply a single fixed regularization coefficient uniformly across all states. This creates a trade-off: in data-rich regions where the model is confident, excessive regularization unnecessarily constrains learning, shifting the policy towards behavior cloning and limiting policy improvement, while in data-sparse regions where the model is uncertain, insufficient regularization fails to prevent value overestimation and leads to extrapolation errors. We propose confidence-based uncertainty regularization enhancement (CURE) that addresses this trade-off using adaptive regularization. CURE aims to tie regularization strength directly to the model's confidence about each state-action pair. CURE quantifies this confidence using *disagreement within a bootstrap ensemble of critics as a measure of epistemic uncertainty*, then feeds this to a learned network that outputs state-specific regularization coefficients. This results in stronger regularization in uncertain regions and optimistic learning where data supports it. CURE, when integrated with established methods from both value regularization and policy constraint paradigms in Offline RL exhibits improvements over baselines on the D4RL benchmark, especially on tasks featuring mixed-quality data.

1 Introduction

Offline reinforcement learning (RL) aims to learn effective policies from previously collected, static datasets. It has a lot of practical value since it bypasses the need for costly or potentially unsafe online environment interaction. However, learning from a fixed dataset introduces an important challenge known as **distributional shift**: the learned policy may query actions that are out-of-distribution (OOD) with respect to the dataset, leading to erroneously high value estimates and subsequent policy degradation [16, 9].

To counter this, state-of-the-art methods typically introduce regularization to prevent the policy from exploiting unreliable value estimates in out-of-distribution regions. This is achieved either by constraining the policy to remain close to the behavior policy that generated the data [8, 21, 17] or by adding a regularization term to the value function objective that penalizes OOD actions [14, 27]. An important limitation of these traditional methods is the use of a *single fixed* regularization coefficient. Fixed regularization coefficients induce a uniform level of conservatism across the entire state space,

*Correspondence to mail.rishav9@gmail.com; † equal supervision

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

leading to a difficult trade-off: Setting this coefficient too high restricts the agent unnecessarily in regions with abundant data, essentially forcing it to just copy the original behavior; while setting it too low fails to protect against errors in regions with sparse data, where the model has little information to rely on.

This limitation has motivated recent work towards adaptive regularization that adjusts strength across different states. Different methods have explored various strategies to achieve this, for example A2PR [17] uses high-advantage actions to guide the regularization strength, PRDC [24] uses dataset proximity constraints to adjust penalties based on distance to nearest dataset samples, and state-adaptive regularization (SSAR) [19] adjusts coefficients based on how closely the policy matches the data distribution and they apply this only on high-advantage actions. While these approaches show promise, they rely on indirect signals or proxy measures rather than directly quantifying the model's uncertainty about its predictions.

Our work, **confidence-based uncertainty regularization enhancement (CURE)**, works on the idea of using the model's own epistemic uncertainty as a direct signal for adaptive regularization. In particular, epistemic uncertainty offers a natural answer to the question *How confident is the model about a given state-action pair?* When multiple critics in an ensemble disagree about a Q-value prediction, this disagreement signals that the query lies outside the model's reliable knowledge, justifying increased conservatism. On the other hand, when critics converge on similar predictions, this agreement indicates high confidence in well-supported regions where the agent can optimize more aggressively.

Building on established ensemble methods for uncertainty estimation [3, 15], CURE implements this idea by using an ensemble of critics in a novel way. Ensembles have been widely used in offline RL: model-based methods [30, 29, 11, 12] use ensembles of dynamics models to estimate uncertainty for safer planning. In the model-free setting, critic ensembles have been used differently. For example, ensemble-diversified actor-critic (EDAC) [1] uses an ensemble of critics to induce implicit pessimism by taking the minimum Q-value across critics while pessimistic bootstrapping for offline RL (PBRL) [2] uses ensemble disagreement as a direct penalty with a fixed coefficient applied uniformly across all states. In contrast, CURE uses the disagreement amongst critics (quantified by the standard deviation of their Q-value predictions) as an explicit signal of epistemic uncertainty. This uncertainty measure is then fed, along with the state, into a small network that learns to output a bounded, state-adaptive regularization coefficient. The result is regularization that automatically adjusts to match the model's confidence at each state.

To validate our approach, we integrate CURE into two established methods representing different offline RL paradigms: conservative Q-learning (CQL) [14] for value function regularization and twin delayed deep deterministic policy gradient + behavior cloning (TD3+BC) [8] for policy constraints. Unlike proxy-based methods, CURE uses direct epistemic uncertainty to scale regularization strength, and integrates easily without requiring architectural changes. We conduct extensive experiments on the datasets for deep data-driven reinforcement learning (D4RL) benchmarks [6], demonstrating that CURE consistently improves performance over the baselines. For example, CURE-TD3+BC achieves 820 points on locomotion tasks compared to the TD3+BC's 679. The gains are particularly pronounced on suboptimal datasets (Table 1).

2 Preliminaries

Markov Decision Processes and Value Functions We consider a standard Markov decision process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \mu_0)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the dynamics or transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and μ_0 is the initial state distribution. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps states to a distribution over actions. The goal of RL is to find an optimal policy π^* that maximizes the expected discounted return. The performance of a policy is measured by its state-action value function (Q-function), which is the expected return after taking action a in state s and following policy π thereafter:

$$Q^\pi(s, a) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

The optimal policy π^* has a corresponding optimal Q-function, $Q^*(s,a) = \max_\pi Q^\pi(s,a)$, which satisfies the Bellman optimality equation:

$$Q^*(s,a) = \mathbb{E}_{s' \sim \mathcal{T}(s,a)} \left[\mathcal{R}(s,a) + \gamma \max_{a' \in \mathcal{A}} Q^*(s',a') \right]$$

Most value-based RL algorithms work by iteratively solving this equation to approximate Q^*.

Offline Reinforcement Learning In offline (or batch) RL, the agent learns exclusively from a static dataset of previously collected transitions, $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$, without any further interaction with the environment. This dataset is typically generated by one or more unknown behavior policies, denoted π_β. The primary challenge in offline RL is distributional shift. Standard online methods, like Q-learning, fail because they must evaluate the Q-values of actions chosen by the new, learned policy. When this policy queries OOD state-action pairs (s', a') that are sparsely or not at all represented in $\mathcal{D}$, the Q-function approximator produces unreliable estimates due to extrapolation error. Because the Bellman update involves a 'max' operator, these errors are systematically exploited, leading to bootstrapped error accumulation and significant overestimation of the Q-values. To combat this, modern offline RL methods introduce conservatism. The goal is to learn a policy that improves upon the behavior policy without moving into OOD regions where value estimates are untrustworthy. This is generally achieved through policy constraints or value function regularization.

Epistemic Uncertainty Uncertainty in machine learning can be decomposed into two types - epistemic and aleatoric uncertainty. Aleatoric uncertainty is irreducible uncertainty inherent in the data-generating process (e.g., a stochastic environment) while epistemic uncertainty, is reducible model uncertainty that stems from a lack of knowledge, typically due to insufficient training data. In offline RL, epistemic uncertainty is a critical signal: it is low for in-distribution state-action pairs that are well-supported by the dataset $\mathcal{D}$ and high for OOD pairs where the Q-function is likely to make extrapolation errors. A common and effective technique for estimating epistemic uncertainty is using a deep ensemble [22], where the disagreement among predictions of networks trained on bootstrapped data serves as a robust estimate of uncertainty. This approach has proven particularly effective in value-based RL [22, 1], enabling algorithms to distinguish between well-supported and uncertain regions of the state space. This uncertainty signal can potentially help an agent to strengthen regularization only when and where the model is uncertain.

3 Related Work

Offline RL learns policies from static datasets making it particularly useful for applications where online interaction is costly or unsafe [25, 4, 16]. The core challenge in offline RL is distributional shift where standard off-policy algorithms fail when querying OOD actions, leading to value overestimation and divergent training [9, 16]. Currently, model-free offline RL is dominated by two main approaches: policy constraints and value function regularization [5].

Policy Constraint Methods Policy constraint methods mitigate distributional shift by keeping the learned policy within the behavior policy's support. This is achieved by through various mechanisms such as explicit policy regularization [9, 7], or by adding direct [8] or advantage-weighted [21] behavior cloning terms to the policy objective. However, a key limitation of traditional policy constraint methods is their use of uniform constraints across all states and actions, which can be overly conservative in data-rich regions while insufficient in data-sparse ones.

Adaptive Policy Constraints. To address this limitation of uniform constraints, recent work has explored adaptive mechanism to adjust the strength of regularization based on local properties. For example, A2PR [17] combines VAE-generated high-advantage actions with adaptive-guided regularization, dynamically selecting constraint targets based on action quality. PRDC [24] uses KD-Tree indexing to identify nearest dataset samples and adjusts constraints based on proximity to the data distribution. While these adaptive approaches show promise, they rely on proxy measures such as action advantage or dataset proximity to determine where regularization is needed.

Value Function Regularization An alternative approach in offline RL, value function regularization, instills conservatism about OOD actions directly into the value function. One of the most popular

works in this domain, CQL, adds a regularizer to learn a lower bound on the policy's value [14]. Other works like implicit Q-learning (IQL) [13], use expectile regression to extract conservative in-distribution value estimates, and soft actor-critic with N Q-ensemble members (SAC-N) [1] extends soft actor-critic with value-based conservatism. However, just like policy constraints methods, these approaches typically apply a single, fixed regularization coefficient uniformly across the entire state space leading to the same problem of excessive conservatism in data-rich regions and insufficient in data-sparse ones.

Adaptive Value Regularization. Recent methods in this paradigm have introduced mechanisms for adaptive value regularization. Adaptive behavior regularization (ABR) [32] adjusts constraint strength based on policy deviation from the behavior policy. exclusively penalized Q-learning (EPQ) [28] selectively applies penalties only to state-action pairs lacking sufficient data support. SSAR [19] learns state-dependent coefficients through a neural network based on policy-data alignment. However, SSAR's effectiveness depends on preprocessing the dataset to filter high-quality actions. In contrast, the proposed CURE uses ensemble-based estimates of epistemic uncertainty to learn state-adaptive regularization coefficients, i.e., it does not require dataset filtering (like SSAR) or algorithmic changes (like A2PR, PRDC), facilitating integration into existing algorithms relying on fixed regularization terms.

Uncertainty in Offline RL Uncertainty estimation has been widely used in offline RL [18] to identify regions where the model's predictions are unreliable. Model-based methods [29, 30, 11, 12, 23] have particularly used it through dynamics model ensembles, using disagreement among ensemble members to identify uncertain state-action pairs and either penalizing them through some objective or avoiding those regions during planning. For example, model-based offline policy optimization (MOPO) [30] subtracts an uncertainty penalty from rewards while model-based offline reinforcement learning (MOReL) [12] constructs a pessimistic MDP based on model disagreement.

In model-free settings, several methods incorporate uncertainty into value function learning. Uncertainty weighted actor-critic (UWAC) [27] uses dropout-based uncertainty to down-weight unreliable Bellman targets. PBRL [2] employs a bootstrapped Q-ensemble and uses the standard deviation of predictions as a penalty term, but applies a fixed coefficient uniformly across all states. EDAC [1] takes the minimum Q-value across an ensemble to induce implicit pessimism, though model standard-deviation gradients (MSG) [10] demonstrates this can paradoxically lead to overestimation when critics share pessimistic training targets. Q-distribution guided Q-learning (QDQ) [31] learns the full Q-value distribution and uses variance as an uncertainty measure but applies global fixed penalties.

A limitation across these uncertainty based methods is the use of fixed penalty coefficients that apply uniformly regardless of local model confidence. In contrast, CURE uses ensemble disagreement to quantify epistemic uncertainty and learns a state-adaptive mapping from this uncertainty to regularization coefficients. This state-adaptive approach enables optimistic learning in well-supported regions while maintaining strong conservatism where the model is uncertain. Importantly, CURE integrates into existing offline RL algorithms as a lightweight module without requiring architectural redesigns or data preprocessing. Figure 1 demonstrates how CURE learns regularization coefficients that directly mirror epistemic uncertainty across the state space, enabling uncertainty-driven adaptive regularization.

4 Critic Ensemble Uncertainty-driven Regularization

Our proposed method, confidence-based uncertainty regularization enhancement (CURE), introduces a framework for learning state-adaptive regularization coefficients in offline RL. The core principle is to use epistemic uncertainty, quantified by the disagreement within a critic ensemble, to dynamically adjust the regularization strength applied to different state-action pairs. This allows the agent to learn optimistically in well-supported regions of the state space while remaining conservative where the model is uncertain. As described in Section 4.4, CURE can be easily integrated with established offline RL algorithms.

4.1 Bootstrap Ensemble for Uncertainty Estimation

To obtain a reliable, non-parametric measure of model uncertainty, CURE uses an ensemble of m independent critic networks, $\{Q_{\theta_i}\}_{i=1}^{m}$. For the ensemble to provide a meaningful uncertainty

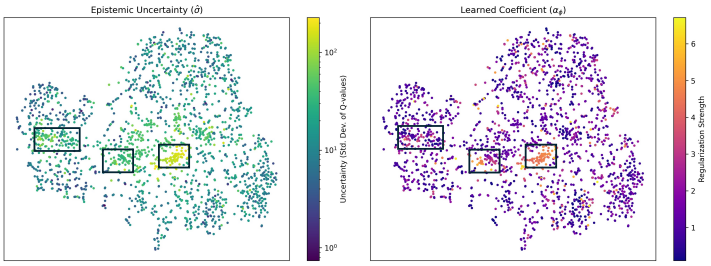


Figure 1: CURE learns a state-dependent regularization coefficient, α_ϕ, that directly mirrors epistemic uncertainty, $\hat{\sigma}$. This figure visualizes this relationship for the `antmaze-umaze-v2` environment using a t-SNE [20] projection of the state space. **(Left)** States are colored by epistemic uncertainty, with high values indicating data-sparse regions where the model is uncertain. **(Right)** The same states are colored by the learned regularization coefficient, α_ϕ. The strong correspondence, with the learned penalty high in uncertain regions and minimal in data-dense areas, provides clear evidence that CURE implements a principled, uncertainty-driven conservative policy. Visualizations for additional environments are included in the supplementary material.

signal, critics must disagree in regions where data is sparse and converge in regions where data is abundant. If all critics see identical training data, they can possibly make similar predictions everywhere, providing no useful signal about which regions are uncertain. To induce this necessary diversity, we use bootstrap sampling: at each training step, every critic Q_{θ_i} is trained on a different bootstrap sample $\mathcal{D}_i$ drawn with replacement from the full offline dataset $\mathcal{D}$. The bootstrap indices are periodically resampled during training to maintain diversity and prevent the ensemble from collapsing to a single solution. We verify the efficacy of bootstrap sampling in our ablation studies in Section 5.2.

Given this diverse ensemble, the epistemic uncertainty for a state-action pair (s, a) is quantified as the standard deviation of the Q-value predictions:

$$\hat{\sigma}(s,a) = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(Q_{\theta_i}(s,a) - \bar{Q}(s,a))^2} \tag{1}$$

where $\bar{Q}(s,a) = \frac{1}{m}\sum_{i=1}^{m} Q_{\theta_i}(s,a)$ is the ensemble mean. This disagreement serves as a direct measure of the model's confidence. High variance indicates high uncertainty, typically corresponding to OOD inputs, while low variance indicates high confidence in regions well-supported by the data.

4.2 State-Adaptive Coefficient Learning

With this direct measure of uncertainty, the most important component of CURE is using this uncertainty signal to dynamically steer regularization strength. Instead of using a fixed regularization coefficient, we introduce a neural network, $\alpha_\phi : \mathcal{S} \times \mathbb{R}^+ \to [\alpha_{\min}, \alpha_{\max}]$, which learns a direct mapping from the current state and its associated uncertainty to a regularization weight. The input to this network is the concatenation of the state vector s and the uncertainty estimate for the policy's action, $u = \hat{\sigma}(s, \pi(s))$. The output is critically constrained to a predefined range $[\alpha_{\min}, \alpha_{\max}]$ to ensure training stability by preventing the regularization term from vanishing or exploding. The mapping is formalized as:

$$\alpha_\phi(s,u) = \alpha_{\min} + \sigma(\text{net}_\phi([s;u])) \cdot (\alpha_{\max} - \alpha_{\min}) \tag{2}$$

where net_ϕ is a multi-layer perceptron with layer normalization, $[s;u]$ denotes concatenation, and $\sigma(\cdot)$ is the sigmoid activation function.

4.3 Multi-Objective Training of the Adaptive Coefficient

Learning an effective adaptive coefficient α_ϕ requires balancing two objectives: responding to local uncertainty signals while maintaining stable training. We train the network using: $\mathcal{L}_\alpha = \mathcal{L}_{\text{calibration}} + \lambda_{\text{stability}} \cdot \mathcal{L}_{\text{stability}}$. The calibration loss encourages α_ϕ to match uncertainty-based targets: $\mathcal{L}_{\text{calibration}} =$

$\mathbb{E}_{s,u}\left[(\alpha_\phi(s,u) - \alpha_{\text{target}}(u))^2\right]$, where the target function maps uncertainty to regularization strength:

$$\alpha_{\text{target}}(u) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \cdot \sigma\left(\frac{u - u_{\text{thresh}}}{u_{\text{thresh}} + \epsilon}\right) \tag{3}$$

The threshold u_{thresh} is set to a high percentile of the current batch's uncertainty distribution, so regularization primarily targets the most uncertain state-action pairs where extrapolation errors are most likely. We study the role of u_{thresh} in our ablation studies in Section 5.2. The sigmoid transformation ensures smooth gradients while operating within the bounds $[\alpha_{\min}, \alpha_{\max}]$. The stability loss prevents the coefficient from changing too much on average: $\mathcal{L}_{\text{stability}} = \left(\mathbb{E}_{s\sim\mathcal{D}}[\alpha_\phi(s,u)] - \bar{\alpha}\right)^2$, where $\bar{\alpha}$ anchors the global behavior. For integration with existing methods, we set $\bar{\alpha}$ to the baseline algorithm's original hyperparameter, providing a starting point while allowing the network to find better strategies. This dual objective is necessary because pure local optimization ($\lambda_{\text{stability}} = 0$) leads to unstable training with rapidly changing regularization, while excessive stability (large $\lambda_{\text{stability}}$) prevents meaningful adaptation to local uncertainty. The balance weight $\lambda_{\text{stability}}$ controls this trade-off. Based on extensive ablation (see Section 5.2), we use $\lambda_{\text{stability}} = 0.1$ in our experiments.

4.4 Integration with Offline RL Algorithms

With this mechanism for learning an adaptive coefficient, CURE can be integrated into offline RL algorithms that use regularization. We demonstrate this on two established methods representing different approaches: policy constraints and value function regularization.

Integration with Policy Constraints. For policy constraint methods, we apply CURE to TD3+BC [8], where regularization takes the form of a behavioral cloning term that constrains the policy to stay close to the dataset actions. We replace its fixed coefficient with our state-adaptive α_ϕ, yielding the actor loss:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_{s\sim\mathcal{D}}[\min_i Q_{\theta_i}(s, \pi_\phi(s))] + \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\alpha_\phi(s, \hat{\sigma}(s, \pi_\phi(s))) \cdot \|\pi_\phi(s) - a\|^2\right] \tag{4}$$

This allows the behavioral cloning constraint to strengthen in uncertain regions where policy deviation is risky, while relaxing in well-supported areas where exploration may be beneficial.

Integration with Value Function Regularization. For value function regularization, we integrate CURE with CQL [14] by adjusting its conservative penalty term with our adaptive coefficient. The loss for each critic Q_{θ_i} becomes:

$$\mathcal{L}_{Q_i} = \mathcal{L}_{\text{Bellman}}(Q_{\theta_i}) + \mathbb{E}_{s\sim\mathcal{D}}\left[\alpha_\phi(s, \hat{\sigma}(s, \pi_\phi(s))) \cdot \mathcal{L}_{\text{CQL}}(Q_{\theta_i}, s)\right] \tag{5}$$

where $\mathcal{L}_{\text{Bellman}}$ is the standard temporal difference (TD) error and $\mathcal{L}_{\text{CQL}}$ is the conservative penalty. This enables CQL to apply stronger pessimism in uncertain states while reducing unnecessary conservatism in well-supported regions.

Both integrations maintain the core algorithmic principles of their baselines while adding uncertainty-aware adaptivity. The critic ensemble is updated using bootstrap samples to maintain diversity, and the uncertainty estimates are computed during training without requiring preprocessing or architectural modifications.

5 Experiments

We conduct a comprehensive set of experiments to evaluate the effectiveness of CURE on D4RL [6] benchmark. Our implementation builds directly upon the publicly available codebase from clean offline reinforcement learning (CORL) [26] to ensure reproducibility and minimize confounding variables arising from implementation details. To facilitate a direct and fair comparison with prior work and to facilitate reproducibility, we adhere strictly to the evaluation protocol established in their study.

5.1 Main Results

Baselines. Our primary evaluation is a direct comparison against the foundational algorithms that CURE enhances: CQL [14] and TD3+BC [8]. We additionally compare against popular methods that employ ensembles and uncertainty in different capacities: EDAC [1], which uses ensemble

Table 1: Average scores across 4 seeds, calculated from the final 10 evaluation runs (10 episodes each; 100 for AntMaze). For baselines (CQL, TD3+BC), we report results with their original ensemble size ($n = 2$) as used in prior work, since our experiments found that increasing to $n = 5$ critics yielded similar or occasionally worse performance for these methods. All CURE-enhanced variants use $n = 5$ critics. For CQL vs C-CQL and TD3+BC vs C-TD3+BC comparisons, we bold scores that are statistically significant improvements ($p < 0.05$). SSAR doesn't report the results on Adroit but we tried running it with their hyper-parameters but the performance was extremely unstable. **Dataset abbreviations:** `hc` (halfcheetah), `hp` (hopper), `wk` (walker2d); `m` (medium), `r` (replay), `e` (expert), `mr` (medium-replay), `me` (medium-expert). **Header abbreviations:** `C-CQL` (CURE-CQL), `C-TD3` (CURE-TD3+BC).

Dataset	CQL	C-CQL	TD3+BC	C-TD3+BC	PBRL	SSAR (CQL)	SSAR (TD3)	EDAC
hc-m-v2	47.03 ± 0.3	$\mathbf{64.1 \pm 1.2}$	48.13 ± 0.7	$\mathbf{65.11 \pm 0.9}$	57.9 ± 1.5	63.9 ± 1.2	56.5 ± 3.7	65.9 ± 0.6
hc-mr-v2	44.2 ± 3.1	$\mathbf{57.32 \pm 0.8}$	45.1 ± 0.8	$\mathbf{58.31 \pm 1.1}$	45.1 ± 8.0	53.8 ± 0.4	49.6 ± 0.3	61.3 ± 1.9
hc-me-v2	95.1 ± 2.3	94.7 ± 1.6	92.9 ± 2.0	91.7 ± 3.9	92.3 ± 1.1	102.1 ± 1.2	94.9 ± 1.2	106.3 ± 1.9
hp-m-v2	62.4 ± 4.8	$\mathbf{92.3 \pm 6.1}$	61.31 ± 3.3	$\mathbf{102.7 \pm 0.7}$	75.3 ± 31.2	89.1 ± 9.7	101.6 ± 0.4	101.6 ± 0.6
hp-mr-v2	67.3 ± 18.19	$\mathbf{103.3 \pm 1.8}$	66.6 ± 27.1	$\mathbf{100.8 \pm 1.1}$	100.6 ± 1.0	101.4 ± 2.1	101.6 ± 0.7	101.0 ± 0.5
hp-me-v2	102.32 ± 5.9	106.3 ± 11.8	91.8 ± 1.2	$\mathbf{105.23 \pm 12.1}$	110.8 ± 0.8	109.6 ± 3.2	103.8 ± 6.7	110.7 ± 0.1
wk-m-v2	82.1 ± 2.2	$\mathbf{89.1 \pm 1.7}$	82.3 ± 2.2	$\mathbf{95.1 \pm 1.1}$	89.6 ± 0.7	84.9 ± 1.7	87.9 ± 2.4	92.5 ± 0.8
wk-mr-v2	81.2 ± 9.4	$\mathbf{95.02 \pm 3.3}$	81.6 ± 7.1	88.23 ± 2.2	77.7 ± 14.5	94.7 ± 3.3	93.5 ± 2.0	87.1 ± 2.3
wk-me-v2	110.56 ± 0.39	$\mathbf{113.56 \pm 1.1}$	110.2 ± 0.4	$\mathbf{113.5 \pm 1.6}$	110.1 ± 0.3	112.2 ± 0.9	112.5 ± 1.4	114.7 ± 0.9
Locomotion Total	692.21	**815.70**	679.94	**820.68**	759.4	811.7	801.9	841.1
antmaze-umaze-v2	92.0 ± 2.2	94.0 ± 3.8	78.6 ± 14.6	88.7 ± 3.3	0 ± 0	96.0 ± 2.3	93.4 ± 3.3	0 ± 0
antmaze-umaze-diverse-v2	31.25 ± 6.75	$\mathbf{68.0 \pm 3.2}$	44.5 ± 16.4	57.0 ± 16.2	0 ± 0	80.2 ± 7.9	50.0 ± 5.4	0 ± 0
antmaze-medium-play-v2	65.0 ± 4.2	73.2 ± 13.3	0 ± 0	$\mathbf{3.0 \pm 0.0}$	0 ± 0	70.2 ± 6.7	49.4 ± 3.4	0 ± 0
antmaze-medium-diverse-v2	62.5 ± 9.2	67.2 ± 10.3	0 ± 0	$\mathbf{4.2 \pm 2.0}$	0 ± 0	71.6 ± 9.3	47.6 ± 12.1	0 ± 0
antmaze-large-play-v2	23.8 ± 6.8	29.2 ± 7.3	0 ± 0	0 ± 0	0 ± 0	53.0 ± 4.1	18.0 ± 4.6	0 ± 0
antmaze-large-diverse-v2	22.2 ± 10.3	19.2 ± 3.3	0 ± 0	0 ± 0	0 ± 0	35.8 ± 18.9	17.6 ± 9.8	0 ± 0
AntMaze Total	296.75	**350.8**	123.1	**152.9**	0 ± 0	406.8	276.0	0 ± 0
pen-human-v1	16.37 ± 7.7	$\mathbf{79.3 \pm 6.9}$	-3.88 ± 0.21	-3.51 ± 0.20	35.4 ± 3.3	-	-	52.1 ± 8.6
pen-cloned-v1	1.04 ± 6.4	$\mathbf{16.13 \pm 2.2}$	5.13 ± 5.28	6.25 ± 5.15	74.9 ± 9.8	-	-	68.2 ± 7.3
pen-expert-v1	0.03 ± 2.33	$\mathbf{99.46 \pm 7.91}$	122.53 ± 21.27	113.15 ± 20.95	137.7 ± 3.4	-	-	106.8 ± 3.4
door-human-v1	5.53 ± 1.31	4.7 ± 3.8	-0.33 ± 0.01	-0.28 ± 0.01	0.1 ± 0.0	-	-	10.7 ± 6.8
door-cloned-v1	-0.33 ± 0.01	$\mathbf{0.9 \pm 2.1}$	-0.34 ± 0.01	-0.29 ± 0.01	4.6 ± 4.8	-	-	9.6 ± 8.3
door-expert-v1	-0.32 ± 0.02	$\mathbf{87.2 \pm 19.3}$	-0.33 ± 0.01	$\mathbf{-0.25 \pm 0.01}$	95.7 ± 12.2	-	-	-
hammer-human-v1	0.14 ± 0.11	$\mathbf{1.02 \pm 1.0}$	1.02 ± 0.24	1.15 ± 0.22	0.4 ± 0.3	-	-	0.8 ± 0.4
hammer-cloned-v1	0.30 ± 0.01	$\mathbf{0.98 \pm 1.8}$	0.25 ± 0.01	$\mathbf{0.35 \pm 0.01}$	0.8 ± 0.5	-	-	0.3 ± 0.0
hammer-expert-v1	0.26 ± 0.01	$\mathbf{82.1 \pm 33.8}$	3.11 ± 0.03	$\mathbf{4.21 \pm 0.03}$	127.5 ± 0.2	-	-	-
relocate-human-v1	0.06 ± 0.03	-0.02 ± 0.22	-0.29 ± 0.01	-0.25 ± 0.01	0.0 ± 0.0	-	-	0.1 ± 0.1
relocate-cloned-v1	-0.29 ± 0.01	$\mathbf{-0.01 \pm 0.18}$	-0.30 ± 0.01	-0.24 ± 0.01	-0.1 ± 0.0	-	-	0.0 ± 0.0
relocate-expert-v1	-0.30 ± 0.0	$\mathbf{67.71 \pm 23.8}$	-1.73 ± 0.96	-1.55 ± 0.90	84.5 ± 12.2	-	-	-
Adroit Total	22.49	**439.47**	124.84	118.99	561.5	-	-	248.6

minimization to induce implicit pessimism through taking minimum Q-values across critics, and PBRL [2], which uses bootstrap ensemble disagreement as a direct penalty but with a fixed coefficient applied uniformly across all states. We include these methods in our table to put CURE's performance in perspective, unlike PBRL or EDAC, CURE is a **not** new algorithmic paradigm but rather acts as a modulator for current methods which use fixed regularization. Additionally, while SSAR [19] is related as an adaptive regularization approach, it is not suitable for direct comparison because it incorporates a selective regularization strategy that applies constraints only to high-quality actions from a filtered subset of the dataset, whereas CURE operates uniformly across the full dataset. In the environments we tested, SSAR without this selective filtering does not show improvement over base algorithms, as reported in their original work and confirmed in our Appendix B. We train all CURE methods for 1 million gradient steps using the publicly available codebase from CORL [26] to ensure reproducibility. For other methods, we utilize numbers reported by them.

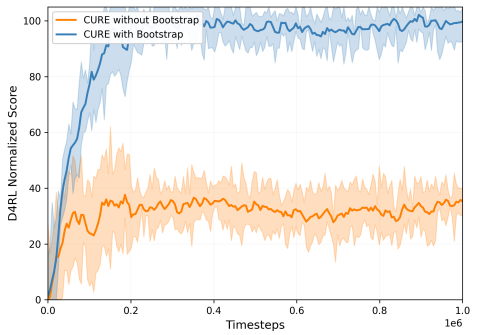
Performance on D4RL [6] Our empirical analysis, presented in Table 1, confirms that CURE provides consistent performance improvements over its foundational baselines. The advantage of our adaptive regularization is clear on aggregate scores; for instance, on the Locomotion benchmarks, CURE-CQL improves upon CQL by 123.49 points (815.70 vs. 692.21), while CURE-TD3+BC surpasses TD3+BC by 140.74 points (820.68 vs. 679.94). This benefit is especially pronounced on challenging mixed-quality datasets. For instance, CURE-TD3+BC achieves a score of 102.7 on `hopper-medium-v2`, a substantial increase from the baseline's 61.31. The efficacy of our approach is also evident in the complex Adroit manipulation suite, where CURE-CQL achieves a total score of 439.47 compared to CQL's 22.49, an improvement of over 416 points.

5.2 Ablation Studies

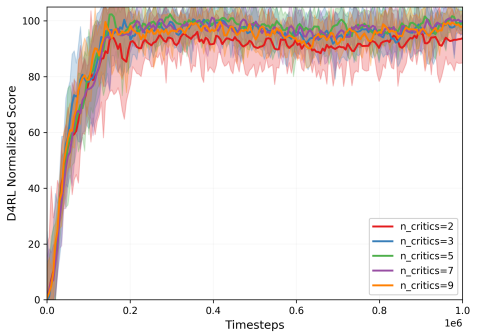
To better understand the contribution of individual components within the CURE framework and to assess its sensitivity to key design choices, we conduct a series of controlled ablation studies. The

majority of these ablations, which analyze the internal mechanics of the CURE module (e.g., the critic ensemble and the coefficient network's training objective), are performed on the CURE-CQL variant for focused analysis. Since these components are designed to be independent of the base algorithm, the insights are broadly applicable. The only exception is the sensitivity to the regularization bounds, $[\alpha_{\min}, \alpha_{\max}]$, which directly interferes with the scale and form of the baseline's loss term. For this reason, we analyze the bounds for both CURE-CQL and CURE-TD3+BC. For these experiments, we primarily utilize the `walker2d-medium-replay-v2` dataset.

Importance of Ensemble Diversity. CURE's uncertainty signal relies on a diverse critic ensemble induced via bootstrap sampling. To validate this, we disabled bootstrapping, forcing all critics to train on identical data. This resulted in significant performance degradation and unstable learning (Figure 2a), demonstrating that data-driven diversity is critical for a reliable uncertainty signal.



(a) Ablation on `walker2d-medium-replay-v2` showing the learning curve of CURE-CQL with and without bootstrap sampling. The removal of bootstrapping leads to significantly degraded performance, highlighting its importance for inducing ensemble diversity and a reliable uncertainty signal.



(b) The effect of varying the number of critics (n) on CURE-CQL's performance in the `walker2d-medium-replay-v2` environment. Performance is robust for smaller ensembles ($n = 2, 3, 5$) and shows a minor decrease for larger ensembles ($n = 7, 9$), suggesting that a small ensemble is sufficient.

Figure 2: Ablation studies for CURE in the `walker2d-medium-replay-v2` environment.

Number of Critics. To assess the sensitivity of CURE to the ensemble size, we evaluated its performance on the `walker2d-medium-replay-v2` environment with a varying number of critics $n \in \{2, 3, 5, 7, 9\}$ for CURE-CQL. As shown in Figure 2b, the results indicate that performance is highly robust for smaller ensembles. The learning curves for $n = 2, 3, 5$ are nearly identical and achieve the highest scores. We observed a slight drop in final performance for the larger ensembles of $n = 7$ and $n = 9$, which could be attributed to potential optimization challenges without further hyperparameter tuning for these larger models. Based on these findings, we selected $n = 5$ as the default for our experiments.

Sensitivity to Regularization Bounds. The adaptive coefficient α_ϕ operates within bounds $[\alpha_{\min}, \alpha_{\max}]$ that serve as training stabilizers. Following the SSAR's [19] convention, we anchor the upper bound to the baseline's validated hyperparameter to provide adaptation headroom while maintaining connection to proven operating regimes. For CURE-CQL, we test $\alpha_{\max} = \kappa \cdot \alpha_{\text{original}}$ with $\kappa \in \{0.2, 1.0, 1.5, 2.0\}$, while for CURE-TD3+BC we use $\kappa \in \{0.4, 1.0, 2.0\}$. We set $\alpha_{\min} = 0.1$ as a small positive lower bound to ensure training stability and prevent numerical issues that arise when regularization terms approach zero.

Empirical validation across both CURE-CQL and CURE-TD3+BC demonstrates robustness to bound specification. Performance remains stable across a wide range of upper bound values for both methods, degrading only when $\alpha_{\max}$ is too restrictive (e.g., $\alpha_{\max} = 1.0$). Even with extreme bounds of $[0, 50]$, both CURE variants show similar improvements over baselines (Table 2), confirming that the uncertainty-driven adaptive mechanism drives performance gains regardless of the underlying algorithm.

Importance of the Dual-Objective for α_ϕ. Our adaptive coefficient network α_ϕ is trained with a dual-component objective combining the calibration loss $\mathcal{L}_{\text{calibration}}$ with a Lagrangian loss $\mathcal{L}_{\text{stability}}$, where the balance is controlled by $\lambda_{\text{stability}}$. To analyze the importance of this balance, we evaluated

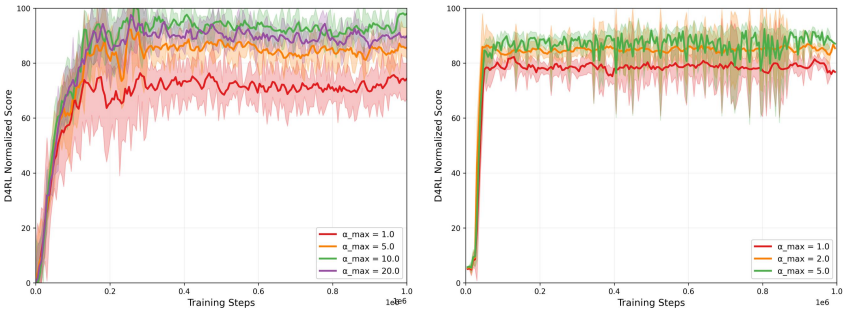


Figure 3: Sensitivity analysis of CURE's regularization bounds, $[\alpha_{\min}, \alpha_{\max}]$. **(Left)** CURE-CQL sensitivity to the upper bound $\alpha_{\max}$ on the `walker2d-medium-replay-v2` dataset. Performance is stable for all tested values $\alpha_{\max} \geq 5.0$, while a lower value of $\alpha_{\max} = 1.0$ leads to slightly degraded performance. **(Right)** CURE-TD3+BC training dynamics on `walker2d-medium-replay-v2`, exhibiting similar sensitivity patterns with $\alpha_{\max} = 5.0$ achieving optimal performance.

performance on the `walker2d-medium-replay-v2` environment while varying $\lambda_{\text{stability}}$ across the values $\{0, 0.1, 0.5, 1.0\}$. The best performance was achieved with $\lambda_{\text{stability}} = 0.1$ (95.5 ± 1.2), while removing the Lagrangian loss entirely ($\lambda_{\text{stability}} = 0$) resulted in unstable dynamics and substantially lower performance (73.0 ± 4.3). Conversely, overweighting the Lagrangian loss ($\lambda_{\text{stability}} = 1.0$) produced a mostly static coefficient that failed to adapt to state-specific uncertainty (81.1 ± 3.5). These findings confirm that both objectives are crucial, with $\lambda_{\text{stability}} = 0.1$ providing a good balance between responsive adaptation and training stability used in our main experiments.

Effect of u_{thresh}. The uncertainty threshold u_{thresh} provides adaptive scaling for the target coefficient α_{target}, addressing the challenge that raw uncertainty values vary significantly across datasets and training phases. Without this scaling, the target function uses $\alpha_{\text{target}}(u) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \cdot \sigma(u)$ where raw uncertainty u is fed directly into the sigmoid, achieving relatively worse performance on `walker2d-medium-replay-v2` (92.1 ± 1.1) for CURE-CQL. Incorporating percentile-based scaling improves performance across different threshold values: $u_{\text{thresh}} \in \{0.9, 0.8, 0.7, 0.6\}$ yield scores of $\mathbf{95.5 \pm 1.5}$, 94.8 ± 1.9, 93.1 ± 2.1, and 93.5 ± 2.4, respectively. This scaling transforms the input to $\sigma\left(\frac{u - u_{\text{thresh}}}{u_{\text{thresh}} + \epsilon}\right)$, converting absolute uncertainty into relative uncertainty within each batch for appropriate distinction between high and low uncertainty regions.

6 Conclusion, Limitations and Future Work

We introduced CURE, a framework that addresses fixed regularization in offline RL by learning a state-adaptive coefficient driven by epistemic uncertainty from a critic ensemble. Experiments on D4RL show that CURE improves performance of strong baselines such as CQLand TD3+BC by applying conservatism precisely where needed, validating the superiority of adaptive and uncertainty-aware regularization over fixed schemes despite modest computational overhead. However, our approach has limitations, including a performance gap on long-horizon, sparse-reward tasks like AntMaze compared with methods that employ data-filtering techniques, and a focus on epistemic uncertainty that may not fully address scenarios where aleatoric uncertainty dominates in highly stochastic environments. Future work could explore alternative uncertainty estimation techniques beyond bootstrap ensembles, such as variational methods or Monte Carlo dropout, investigate CURE's effectiveness in stochastic environments where aleatoric uncertainty is significant, extend this principle to other paradigms such as model-based reinforcement learning (MBRL).

References

[1] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified Q-ensemble. In *Advances in Neural Information Processing Systems*, volume 34, pages 7436–7447, 2021.

[2] Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhihong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning.

In *International Conference on Learning Representations*, 2022.

[3] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[4] Christopher Diehl, Timo Sievernich, Martin Krüger, Frank Hoffmann, and Torsten Bertram. Umbrella: Uncertainty-aware model-based offline reinforcement learning leveraging planning. *arXiv preprint arXiv:2111.11097*, 2021.

[5] Rafael Figueiredo Prudencio, Marcos R. O. A. Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):10237–10257, 2024.

[6] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021.

[7] Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms, 2019.

[8] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning, 2021.

[9] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. *International Conference on Machine Learning*, 2019.

[10] Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline RL through ensembles, and why their independence matters. *arXiv preprint arXiv:2205.13703*, 2022.

[11] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization, 2021.

[12] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel : Model-based offline reinforcement learning, 2021.

[13] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning, 2021.

[14] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

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

[16] 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.

[17] Tenglong Liu, Yang Li, Yixing Lan, Hao Gao, Wei Pan, and Xin Xu. Adaptive advantage-guided policy regularization for offline reinforcement learning, 2024.

[18] Owen Lockwood and Mei Si. A review of uncertainty for deep reinforcement learning, 2022.

[19] Qin-Wen Luo, Ming-Kun Xie, Ye-Wen Wang, and Sheng-Jun Huang. Learning to trust bellman updates: Selective state-adaptive regularization for offline rl. *arXiv preprint arXiv:2505.19923*, 2025.

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[21] Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[22] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[23] Zhongjian Qiao, Jiafei Lyu, Kechen Jiao, Qi Liu, and Xiu Li. Sumo: Search-based uncertainty estimation for model-based offline reinforcement learning, 2024.

[24] Yuhang Ran, Yi-Chen Li, Fuxiang Zhang, Zongzhang Zhang, and Yang Yu. Policy regularization with dataset constraint for offline reinforcement learning, 2023.

[25] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[26] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. Corl: Research-oriented deep offline reinforcement learning library. *Advances in Neural Information Processing Systems*, 36:30997–31020, 2023.

[27] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*, 2021.

[28] Junghyuk Yeom, Yonghyeon Jo, Jeongmo Kim, Sanghyeon Lee, and Seungyul Han. Exclusively penalized q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:113405–113435, 2024.

[29] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization, 2022.

[30] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization, 2020.

[31] Jing Zhang, Linjiajie Fang, Kexin Shi, Wenjia Wang, and Bing-Yi Jing. Q-distribution guided q-learning for offline reinforcement learning: Uncertainty penalized q-value via consistency model, 2025.

[32] Yunfan Zhou, Xijun Li, and Qingyu Qu. Offline reinforcement learning with adaptive behavior regularization. In *International Conference on Learning Representations*, 2023.

A More on Regularization Bounds

CURE demonstrates robustness to its regularization bounds $[\alpha_{\min}, \alpha_{\max}]$. Even with extremely wide bounds of $[0, 50]$, our method achieves significant improvements over baselines, confirming that substantial performance gains do not depend on precise hyperparameter tuning. While the bounds specified in our main experiments ($\alpha_{\min} = 0.5$ for medium-expert datasets) yield optimal results, CURE's effectiveness primarily derives from its uncertainty-driven adaptive mechanism rather than careful bound selection.

Table 2: Performance with regularization bounds $[\alpha_{\min}, \alpha_{\max}] = [0, 50]$

Environment	C-CQL	C-TD3+BC
hc-medium-v2	60.3 ± 2.2	62.1 ± 3.1
hp-medium-replay-v2	101.3 ± 2.2	100.8 ± 2.2
hp-medium-expert-v2	104.3 ± 13.1	95.71 ± 17.54
wk-medium-expert-v2	110.1 ± 1.2	110.4 ± 2.5
hc-medium-expert-v2	92.8 ± 2.1	90.4 ± 1.2

B Baseline: SSAR [19] without filtering

As previously discussed in both SSAR and our main paper, the effectiveness of SSAR diminishes markedly when filtering is omitted. For direct comparison, we report empirical results of SSAR without filtering across several environments. Notably, SSAR without filtering exhibits a pronounced reduction in performance, achieving results that are on par with baseline algorithms. The following table summarizes these findings.

Table 3: Performance degradation of SSAR when selective filtering is removed, compared to CURE-CQL on selected D4RL environments. Results demonstrate SSAR's dependence on data preprocessing, with performance below CURE-CQL when operating on the full dataset.

Environment	SSAR-no-filter-CQL	C-CQL
halfcheetah-medium-v2	48.1 ± 0.4	64.1 ± 1.2
hopper-medium-v2	66.3 ± 8.2	92.3 ± 6.1
walker2d-medium-v2	81.7 ± 1.2	89.1 ± 1.7
walker2d-medium-replay-v2	83.3 ± 6.3	95.02 ± 3.3
hopper-medium-replay-v2	92.3 ± 3.3	103.3 ± 1.8
walker2d-medium–v2	108.3 ± 2.7	113.56 ± 1.1

C Uncertainty vs Regularization t-SNE

These are additional t-SNE visualizations demonstrating the correspondence between epistemic uncertainty and regularization strength across different environments (see Figure 4).

C.1 Adaptive Regularization Network Architecture

The adaptive regularization coefficient network is a core component of CURE that learns to map state-uncertainty pairs to regularization coefficients. The network takes as input the concatenation of the current state $s \in \mathbb{R}^{d_s}$ and the epistemic uncertainty estimate $\hat{\sigma}(s, a) \in \mathbb{R}$ for the policy's action at that state, producing a bounded regularization coefficient $\alpha_\phi(s, \hat{\sigma}) \in [\alpha_{\min}, \alpha_{\max}]$.

The network architecture consists of:

- **Input Layer:** Layer normalization applied to the concatenated state-uncertainty vector of dimension $(d_s + 1)$
- **Hidden Layers:** Two fully connected layers of 256 units each with ReLU activations
- **Output Layer:** Single linear layer followed by sigmoid activation to ensure bounded output

The final output is computed as:

$$\alpha_\phi(s, \hat{\sigma}) = \alpha_{\min} + \sigma(\text{net}_\phi([s; \hat{\sigma}])) \cdot (\alpha_{\max} - \alpha_{\min}) \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function, net_ϕ represents the neural network, and $[s; \hat{\sigma}]$ denotes concatenation. The network is initialized using orthogonal initialization for improved training stability.

The network is trained using a dual-objective loss function that balances calibration to uncertainty-based targets with a Lagrangian regularizer to maintain stable global behavior, as described in the main paper.

D Computational Cost

The CURE framework is designed for high computational efficiency, stemming from two key factors. First, the adaptive regularization network is lightweight and imposes a minimal computational burden. Second, the critic ensemble architecture is highly parallelizable, allowing modern GPUs to process numerous critics with little additional latency. Consequently, as demonstrated in Table 4, the runtime overhead of introducing CURE is negligible compared to the baselines.

E Codebase

Our implementation builds upon the codebase from CORL [26]. To ensure reproducibility, we open-source our complete implementation at `https://github.com/rish-av/cure-offline-rl`.

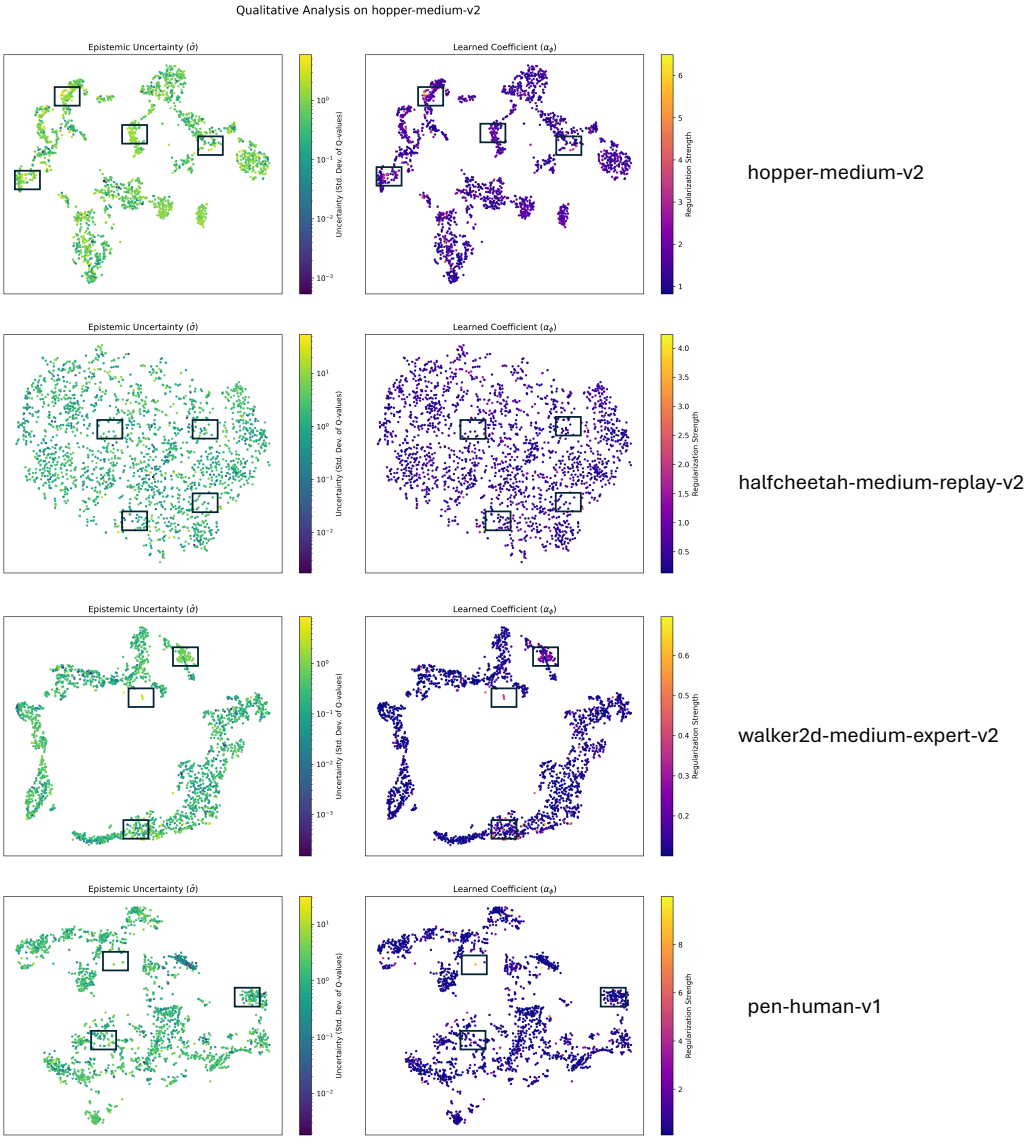


Figure 4: Additional qualitative analysis showing CURE's state-dependent regularization coefficient α_ϕ directly mirroring epistemic uncertainty $\hat{\sigma}$ across four additional D4RL environments using t-SNE projections of the state space. For each environment (hopper-medium-v2, halfcheetah-medium-replay-v2, walker2d-medium-expert-v2, and pen-human-v1), the left panel shows states colored by epistemic uncertainty, with high values indicating data-sparse regions where the model is uncertain. The right panel shows the same states colored by the learned regularization coefficient α_ϕ. Consistent with the main results, the strong correspondence across all environments demonstrates that CURE successfully implements principled, uncertainty-driven conservative policies, with learned penalties high in uncertain regions and minimal in data-dense areas.

Table 4: Runtimes per epoch on an NVIDIA H100 GPU. CURE introduces negligible overhead over baselines and scales efficiently as the number of critics (n) increases.

Algorithm	**Critics (n)**	**Runtime (s)**
CQL (baseline)	2 (default)	12.8
CURE-CQL	2	13.0
CURE-CQL	9	13.4
TD3+BC (baseline)	2 (default)	2.6
CURE-TD3+BC	2	2.7
CURE-TD3+BC	9	2.7

Table 5: CURE-specific hyperparameters for both CQL and TD3+BC implementations. Standard CQL hyperparameters follow `https://github.com/corl-team/CORL/blob/main/algorithms/offline/cql.py` and standard TD3+BC hyperparameters follow `https://github.com/corl-team/CORL/blob/main/algorithms/offline/td3_bc.py`.

Hyperparameter	CURE-CQL	CURE-TD3+BC	Description
`use_cure`	True	True	Enable state-adaptive regularization
`cure_target_penalty`	15.0	-	Target penalty value for Lagrangian loss
`cure_alpha_lr`	5e-5	5e-5	Learning rate for adaptive alpha network
`cure_warmup_steps`	0	0	Steps before enabling CURE (uses fixed penalty)
`min_alpha`	0.1	0.1	Minimum regularization coefficient
`max_alpha`	20.0	5.0	Maximum regularization coefficient
`n_critics`	5	5	Number of critics in ensemble
`bootstrap_resample_freq`	5000	5000	Frequency of bootstrap indices resampling