CURE: Conservative Unlearning with Soft-Gating Regularization for Offline Reinforcement Learning

Bijal Bharadva[†], **Manuel Kelly**[†], **Yingxue Zhang**[$], **Xin Zhang**[†*]
San Diego State University, USA[†], Binghamton University, USA[$]
{bbharadva1752, mkelly5736}@sdsu.edu,
yzhang42@binghamton.edu, xzhang19@sdsu.edu

Abstract

Offline reinforcement learning (RL) trains policies from static datasets, but deployed agents may later need to attenuate the influence of specific data for privacy, safety, or fairness. This is challenging in the overlap regime, where the forget set and retained data share state-action support: unlearning cannot be uniform as suppressing values on shared support can inadvertently erase behaviors required for high performance. We propose Conservative Unlearning with soft-gating Regularization (CURE), a selective unlearning framework tailored to overlap. CURE assigns each forget trajectory an overlap-interference score via cosine alignment between its temporal difference (TD) semi-gradient and a retained reference gradient, and uses it to soft-gate *i)* conservative, state-conditional critic suppression relative to a retained baseline and *ii)* coupled actor updates, selectively modulating forgetting based on interference, reducing impact on aligned data. We provide a first-order analysis showing that CURE descends a forget surrogate while controlling retained drift via alignment-dependent coupling. Across MuJoCo benchmarks and offline RL backbones, CURE consistently improves the forget-utility trade-off over baselines, achieving low auditor-detectable forget influence with near-original returns at a fraction of retraining cost.

1 Introduction

Offline reinforcement learning (RL) [10, 13, 12] enables learning decision-making policies from previously collected datasets, supporting deployment in domains where online interaction is costly, risky, or infeasible *e.g.*, autonomous driving [4], healthcare [18], and robotics [16]. As offline RL systems move closer to practice, an important requirement emerges: machine unlearning [2]. Concretely, given an offline dataset and a designated forget subset, machine unlearning aims to remove the influence of forget data without the prohibitive cost of retraining, while maintaining performance. In offline RL, such requests arise from privacy and governance requirements (*e.g.*, the GDPR "right-to-be-forgotten"), safety needs (*e.g.*, deleting logged episodes corresponding to unsafe driving maneuvers identified in safety audits), and post-hoc correction of deployed policies.

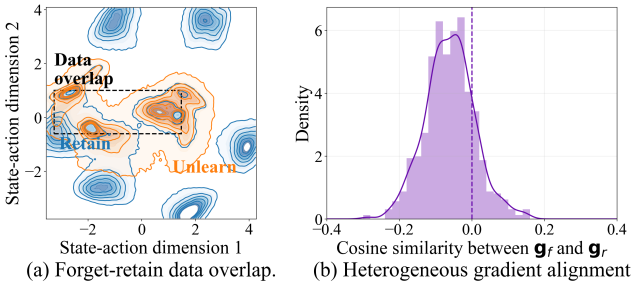
Challenges. Unlearning in offline RL is fundamentally subtle: *i) Bootstrapped targets:* Unlike supervised unlearning with fixed labels, offline RL optimizes bootstrapped value targets that depend on the current critic and policy, so edits to unlearn forget data can propagate beyond the forget samples. *ii) Data overlap:* The forget and retained data often share substantial state-action support (Fig. 1(a)), so deleting trajectories does not isolate their effects: even retraining on the retained data can reproduce behaviors correlated with the forget set in shared regions, leaving detectable forget-supported influence. *iii) Actor-critic coupling:* Value and policy learning are tightly coupled, so local critic changes can alter actor updates and amplify unlearning into broader behavior shifts.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Consequently, the standard unlearning goal of "matching retraining on the retained data" can be insufficient, while uniformly penalizing the forget set can over-suppress shared behaviors and degrade utility. This creates a central tension: *How can we remove forget data influence with minimal retained performance loss under state-action overlap?*

Limitations of existing approaches. These challenges also explain why conventional machine unlearning methods [2, 20, 1] do not transfer cleanly to offline RL. Many are developed for supervised objectives with sample-decomposable losses and relatively stable gradients, whereas offline RL objectives are bootstrapped and policy-dependent, and actor-critic feedback can amplify local edits into global behavior changes. Meanwhile, recent RL unlearning works [8, 21, 9] often implement forgetting by suppressing critic values on the forget set while fine-tuning on the remaining data. Under overlap, uniform suppression can be hard to calibrate: it treats all forget trajectories equally regardless of whether they conflict with retained learning signals, potentially leading to over-suppression in shared regions or weaker forget-utility tradeoffs. What is missing is a mechanism to prioritize which forget trajectories truly interfere with retained learning in shared regions.



(a) Forget-retain data overlap. (b) Heterogeneous gradient alignment

Figure 1: (a) State-action overlap between forget and retained data (2D t-SNE with KDE). (b) Cosine similarity between each forget trajectory's gradient and a retained reference gradient: many forget gradients align with retained learning.

Motivation. Under overlap, unlearning should not be interpreted as uniformly erasing all behaviors associated with the forget set, since shared state-action support makes such erasure incompatible with preserving retained utility. Instead, we study overlap-aware offline RL unlearning, where the objective is to attenuate forget-side optimization influence that conflicts with retained learning, while limiting collateral drift on retained-supported behavior. In this sense, conservative unlearning under overlap means *minimum-necessary suppression*, rather than uniform suppression of all forget data. This reframes unlearning as an *interference* problem rather than a *membership* problem: the key question is not only whether a trajectory belongs to the forget set, but whether its learning signal conflicts with retained learning. In offline RL, that influence is expressed in the temporal difference (TD) semi-gradient, which captures the direction of critic updates. We therefore measure the *overlap interference* of a forget trajectory by cosine alignment between its TD-gradient and a retained reference TD-gradient. Importantly, gradient alignment serves as a proxy for interference under unlearning: higher alignment implies stronger coupling with retained learning and thus larger retained-side interference when suppressed, whereas weakly aligned trajectories incur minimal damage. Empirically, we observe substantial heterogeneity in alignment (Fig. 1(b)), indicating that uniform penalization can induce over-forgetting by suppressing directions needed for retained utility.

Our approach. We propose CURE, *i.e.*, Conservative Unlearning with soft-gating Regularization, an offline RL unlearning framework tailored to the overlap regime. Concretely, CURE assigns each forget trajectory a *trajectory-level overlap-interference score* using cosine alignment between its TD semi-gradient and a retained reference gradient, and uses this score to soft-gate critic and actor unlearning strength. CURE performs *state-conditional* critic editing anchored by a retained baseline: it suppresses forget-induced values only when they exceed what is justified by retained data in the same state, thereby avoiding indiscriminate value suppression on shared support. Finally, CURE uses the same gate to couple actor updates, preventing actor-critic feedback from amplifying localized critic edits into global policy drift. While gradient-similarity heuristics exist broadly, CURE 's combination of TD-gradient-based interference scoring, retained-baseline conservative suppression under overlap, and shared-gate actor-critic unlearning is specific to offline RL unlearning with bootstrapping and state-action overlap. *Our contributions are as follows* [2]*:*

- We formalize overlap-aware offline RL unlearning under shared state-action support, where retrain-matching and uniform suppression can both be inadequate.

[2]Code is available at `https://github.com/STAI-SDSU/CURe`.

- We propose CURE, which scores forget trajectories by TD-gradient interference and uses a shared soft gate to couple critic suppression with actor forgetting as a plug-in across offline RL backbones.
- We show CURE descends a forget surrogate while controlling retained drift via alignment-dependent terms.
- We empirically show that CURE improves the forget-utility trade-off under overlap, achieving low auditor-detectable forget influence, strong retained utility, and substantial efficiency gains over retraining and prior baselines on MuJoCo tasks [19].

2 Preliminaries

Offline reinforcement learning. We consider a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$ and an offline dataset $\mathcal{D} = \{\tau_i\}_{i=1}^N$ of trajectories, where each trajectory $\tau = \{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^T$ is collected by an unknown behavior policy. Offline RL aims to learn a policy π maximizing expected discounted return $J(\pi) = \mathbb{E}_{\pi,P}\big[\sum_{t=1}^T \gamma^{t-1} r(s_t, a_t)\big]$ using only $\mathcal{D}$.

Offline RL unlearning under overlap. Let $\mathcal{D}_f \subset \mathcal{D}$ denote a *forget set* of trajectories designated for unlearning, and $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ denote the *retained data*. We focus on the *overlap regime*, where $\mathcal{D}_f$ and $\mathcal{D}_r$ share state-action support. Let $\mathcal{A}(\cdot)$ be an offline RL algorithm producing a policy $\pi = \mathcal{A}(\mathcal{D})$. An unlearning algorithm $\mathcal{U}(\cdot)$ takes $(\pi, \mathcal{D}, \mathcal{D}_f)$ and outputs an updated policy $\pi' = \mathcal{U}(\pi, \mathcal{D}, \mathcal{D}_f)$ at substantially lower cost than retraining. We seek π' that: *i) Efficacy:* removes detectable influence of $\mathcal{D}_f$, as assessed by auditors (*e.g.*, TrajAuditor [8]) that test whether π' still reflects information from $\mathcal{D}_f$; *ii) Utility:* preserves task performance; *iii) Efficiency:* achieves *i)-ii)* with lower computation than retraining from scratch. Under overlap, we therefore do not define successful unlearning as exact removal of every behavior appearing in the forget set. Instead, we evaluate unlearning through the forget-utility trade-off: reducing detectable forget-supported influence while preserving performance on behavior still supported by retained data.

3 Conservative Unlearning with Soft-Gating Regularization

We now instantiate CURE (Conservative Unlearning with soft-gating Regularization), a conservative unlearning framework for offline RL under shared state-action support. CURE addresses challenges in Sec. 2 via: *i)* temporal difference (TD)-gradient-based interference score that measures each forget trajectory's conflict with the retained learning signal under bootstrapping; *ii)* overlap-aware, state-conditional suppression that penalizes forget actions relative to a retained baseline at the same state; and *iii)* coupled actor-critic unlearning that applies the same interference gate to actor-critic updates, preventing critic-only edits from resurfacing.

TD-based interference estimation. Given a trajectory τ, define its critic TD regression loss $\ell_{\mathrm{TD}}(\tau;\phi) = \frac{1}{T}\sum_{t=1}^T \big(Q_\phi(s_t,a_t) - y_t\big)^2$, where Q_ϕ is the critic, and the TD target $y_t = r_t + \gamma \mathbb{E}_{a'\sim\pi(\cdot|s_{t+1})} Q_{\bar{\phi}}(s_{t+1}, a')$ is treated as a constant *w.r.t.* ϕ (semi-gradient TD) when $Q_{\bar{\phi}}$ is a target critic. To quantify how strongly a forget trajectory interferes with retained learning, we compare its TD gradient with a retained reference. For $\forall \tau' \in \mathcal{D}_f$, let $\mathbf{g}_f(\tau') := \nabla_\phi \ell_{\mathrm{TD}}(\tau';\phi)$, and a retained reference gradient $\mathbf{g}_r := \mathbb{E}_{\tau\sim\mathcal{D}_r}\big[\nabla_\phi \ell_{\mathrm{TD}}(\tau;\phi)\big]$. We compute the cosine similarity $\mathrm{sim}(\tau')$, and map it to a bounded *forget gate* $w(\tau')$ defined below:

$$\mathrm{sim}(\tau') := \frac{\langle \mathbf{g}_f(\tau'), \mathbf{g}_r\rangle}{\|\mathbf{g}_f(\tau')\|\|\mathbf{g}_r\|} \in [-1,1], \qquad w(\tau') := \frac{1-\mathrm{sim}(\tau')}{2} \in [0,1]. \tag{1}$$

Aligned trajectories ($\mathrm{sim}(\tau') \approx 1$) receive small gates enabling conservative updates, while conflicting ones ($\mathrm{sim}(\tau') < 0$) receive larger gates and are emphasized for forgetting. Unlike supervised reweighting, this gate is defined *w.r.t.* the bootstrapped retained TD signal and is used to modulate coupled actor-critic edits under overlap. In practice, $\mathbf{g}_r$ is estimated from a retained minibatch. To control overhead, $\mathbf{g}_f(\tau')$ is computed per grouped mini-trajectory batch. We treat $w(\tau')$ as a constant during unlearning to avoid gradients through cosine similarity.

Interference-gated conservative suppression. CURE addresses over-forgetting under overlap with *state-conditional suppression*, so forget actions are penalized relative to retained behavior at the same state, and *interference-gated suppression*, so aligned forget data are modified minimally.

A retained baseline value $V_r(s) = \mathbb{E}_{a\sim\beta_r(\cdot|s)}[Q_\phi(s,a)]$ is defined, where $\beta_r(\cdot|s)$ is the empirical retained action distribution in $\mathcal{D}_r$ that is estimated by minibatch sampling[3]. We penalize forget actions using a hinge $[Q_\phi(s,a) - V_r(s)]_+$ with $[x]_+ := \max(x, 0)$. It penalizes a forget action only when its value exceeds the retained baseline, which avoids forcing Q_ϕ toward a fixed minimum and reduces collateral suppression at overlapping states.

Since different forget trajectories have varied interferences on retained learning, we gate the hinge penalty based on τ''s interference and introduce the CURE suppression term,

$$\mathcal{R}_{\text{CURe}}(\tau') = w(\tau') \cdot \frac{1}{T} \sum_{t=1}^{T} \big[Q_\phi(s_t, a_t) - V_r(s_t)\big]_+.$$

Coupled actor-critic unlearning. Given an offline RL backbone with critic loss $\mathcal{L}_Q(\mathcal{D}_r; \phi)$ on $\mathcal{D}_r$ (*e.g.*, critic regression in IQL [10] with conservative regularizers in CQL [12]), CURE augments the critic update with gated suppression:

$$\mathcal{L}_Q^{\text{CURe}}(\phi) = \underbrace{\mathcal{L}_Q(\mathcal{D}_r; \phi)}_{\text{Retain Critic Fit}} + \lambda \underbrace{\mathbb{E}_{\tau'\sim\mathcal{D}_f}\Big[\mathcal{R}_{\text{CURe}}(\tau')\Big]}_{\text{CURe Critic Suppression}}, \tag{2}$$

where λ controls forgetting strength. The suppression term removes value incentives for forget actions state-conditionally: if (s, a) overlaps retained support with high $V_r(s)$, it reduces over-forgetting shared behaviors.

Since offline actor-critic learning is coupled, critic-only edits can reintroduce forget behaviors through policy optimization. For actor-critic backbones, CURE reuses the critic-based gate $w(\tau')$ for actor forgetting. Given a backbone actor loss $\mathcal{L}_\pi(\mathcal{D}_r; \theta)$ on $\mathcal{D}_r$, CURE minimizes:

$$\mathcal{L}_\pi^{\text{CURe}}(\theta) = \mathcal{L}_\pi(\mathcal{D}_r; \theta) + \eta \underbrace{\mathbb{E}_{\tau'\sim\mathcal{D}_f}\Big[w(\tau') \cdot \ell_{\text{sup}}(\tau'; \theta)\Big]}_{\text{CURe Actor Suppression}}, \tag{3}$$

where η controls actor forgetting strength, and ℓ_{sup} is a backbone-specific suppression term that discourages the policy from reproducing forget actions at forget states. For explicit stochastic actors, $\ell_{\text{sup}}(\tau'; \theta) = \frac{1}{T}\sum_{t=1}^{T} \log \pi_\theta(a_t|s_t)$; other backbone parameterizations are adapted accordingly in Appx. D. In all cases, the suppression discourages reproduction of forget actions while the gate $w(\tau')$ conservatively protects aligned data. The complete procedure is summarized in Alg. 1.

4 Interference Gating & Utility Drift Analysis

We analyze CURE as an interference-gated update under overlap, focusing on how alignment-based gating modulates the coupling between forget-side suppression and retained learning. We use first-order optimization tools to study one-step behavior under mild smoothness assumptions, treating the gate $w(\tau')$ as fixed within each update step.

4.1 Setup: Smooth Hinge Relaxation

Smooth approximation of the hinge. Since the hinge operator $[x]_+$ is non-smooth at $x = 0$, we use a smooth approximation when analyzing Eq. (2): $\rho_\beta(x) = \beta \log\big(1 + \exp(\frac{x}{\beta})\big)$, with small $\beta > 0$.

Interference-gated forget surrogate. Let $\psi(s, a; \phi) := Q_\phi(s, a) - V_r(s)$, where the retained baseline $V_r(s)$ is treated as fixed within one update. We define the interference-gated forget surrogate, $\Psi(\phi) := \mathbb{E}_{\tau'\sim\mathcal{D}_f}\big[w(\tau') \cdot \frac{1}{T}\sum_{t=1}^{T} \rho_\beta\big(\psi(s_t, a_t; \phi)\big)\big]$, which corresponds to CURE 's critic suppression in Eq. (2)[4]. For compactness, when $\beta \to 0$, we rewrite the CURE critic objective as, $\mathcal{L}_Q^{\text{CURe}}(\phi) = \mathcal{L}_Q(\mathcal{D}_r; \phi) + \lambda\, \Psi(\phi)$.

[3] In continuous-control settings, we approximate $V_r(s)$ using retained actions $\tilde{a}$ associated with s when available, or with nearby retained states otherwise; concretely, we use a minibatch-local nearest-neighbor approximation. Details are in Appx. C.

[4] $\rho_\beta(x)$ approximation satisfies $[x]_+ \le \rho_\beta(x) \le [x]_+ + \beta \log 2$, so $\Psi(\phi)$ upper bounds the gated hinge violation up to an additive $\beta \log 2$ per transition, which vanishes as $\beta \to 0$.

Smoothness and gradients. Assume $\mathcal{L}_Q(\cdot)$ and $\Psi(\cdot)$ are L-smooth in ϕ on a bounded parameter set. We present a single full-gradient update; in practice CURE uses minibatch gradients, and the same bounds hold in expectation under standard unbiasedness and bounded-variance assumptions.

4.2 One-step Forgetting and Retained Drift

We first characterize how one CURE critic step changes the forget surrogate (Prop. 1) and the retained critic objective (Prop. 2) when the gate is treated as fixed within the update. The key point is that both forgetting progress and retained drift are governed by the same coupling term.

Proposition 1 (One-step change of the forget surrogate)**.** *Consider one* CURE *critic update* $\phi^+ = \phi - \alpha\Big(\nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi) + \lambda \nabla_\phi \Psi(\phi)\Big)$*, with* $\alpha \le \frac{1}{L}$*, the forget surrogate satisfies*

$$\Psi(\phi^+) \le \Psi(\phi) - \alpha\lambda \big\|\nabla_\phi \Psi(\phi)\big\|^2 - \alpha \big\langle \nabla_\phi \Psi(\phi), \nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi)\big\rangle + \frac{L\alpha^2}{2}\Big\|\nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi) + \lambda \nabla_\phi \Psi(\phi)\Big\|^2.$$

Proof sketch. By L-smoothness of Ψ [3], $\Psi(\phi^+) \le \Psi(\phi) + \langle \nabla\Psi(\phi), \phi^+ - \phi\rangle + \frac{L}{2}\|\phi^+ - \phi\|^2$. Substitute $\phi^+ - \phi = -\alpha(\nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi) + \lambda\nabla_\phi\Psi(\phi))$ and expand. See Appx. A.2. □

Prop. 1 quantifies one-step progress on the forget surrogate. Up to $\mathcal{O}(\alpha^2)$ terms, forgetting progress is governed by a descent term $-\alpha\lambda\|\nabla_\phi\Psi(\phi)\|^2$ and a coupling term $\big\langle \nabla_\phi \Psi(\phi), \nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi)\big\rangle$ that captures overlap-induced interference from retained learning. The same coupling term also governs how forgetting perturbs the retained critic objective.

Proposition 2 (One-step retained critic drift bound)**.** *Under the same update as Prop. 1 and step size* $\alpha \le 1/L$*, the change of retained backbone critic loss satisfies*

$$\begin{aligned}\mathcal{L}_Q(\phi^+) \le \mathcal{L}_Q(\phi) &- \alpha\big\|\nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi)\big\|^2 - \alpha\lambda\big\langle \nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi), \nabla_\phi \Psi(\phi)\big\rangle \\ &+ \frac{L\alpha^2}{2}\Big\|\nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi) + \lambda \nabla_\phi \Psi(\phi)\Big\|^2.\end{aligned}$$

Proof sketch. Apply L-smoothness of $\mathcal{L}_Q(\cdot)$ leads to the result as in Prop. 1. See Appx. A.3. □

Prop. 1-2 show that forgetting progress and retained drift arise from the same overlap-induced tension: the more forget-side suppression conflicts with retained learning, the more retained drift is incurred.

4.3 Alignment-gated Control of Coupling

Propositions 1-2 identify the coupling term as the central quantity governing both forgetting progress and retained drift. To make the role of CURE's gate explicit in controlling this coupling, we decompose the forget-side gradient into per-trajectory contributions. Define the per-trajectory gradient contribution to the forget surrogate $\mathbf{g}_\Psi(\tau') := \frac{1}{T}\sum_{t=1}^{T} \nabla_\phi \rho_\beta\Big(\psi(s_t, a_t;\phi)\Big)$, so that $\nabla_\phi \Psi(\phi) = \mathbb{E}_{\tau'\sim\mathcal{D}_f}\big[w(\tau')\cdot \mathbf{g}_\Psi(\tau')\big]$. Then the coupling term can be rewritten as,

$$\big\langle \nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi), \nabla_\phi \Psi(\phi)\big\rangle = \|\mathbf{g}_r\| \cdot \mathbb{E}_{\tau'\sim\mathcal{D}_f}\Big[w(\tau')\cdot\|\mathbf{g}_\Psi(\tau')\|\cdot\cos(\tau')\Big], \tag{4}$$

where $\cos(\tau') = \frac{\langle \mathbf{g}_\Psi(\tau'), \mathbf{g}_r\rangle}{\|\mathbf{g}_\Psi(\tau')\|\,\|\mathbf{g}_r\|}$ is the cosine alignment between the trajectory's forget suppression gradient and the retained critic gradient. Eq. (4) shows that trajectories that are more aligned with retained learning contribute less when assigned smaller gates, thereby reducing the effective coupling that drives retained drift. Although $w(\tau')$ is computed from TD gradients, both TD and suppression gradients share the factor $\nabla_\phi Q_\phi$, so TD alignment serves as a lightweight proxy for interference with the conservative suppression direction.

Corollary 3 (Interference gating mitigates interference)**.** CURE *sets* $w(\tau')$ *as Eq. (1), which is non-increasing in the TD-gradient alignment* $\mathrm{sim}(\tau')$*. Therefore, trajectories whose learning signals align with retained learning receive smaller gates. Combined with Eq. (4), this implies that when TD-aligned trajectories dominate the forget set,* CURE *reduces the effective coupling term* $\big\langle \nabla_\phi \mathcal{L}_Q(\mathcal{D}_r;\phi), \nabla_\phi \Psi(\phi)\big\rangle$*, mitigating coupling-driven retained drift for a given* λ*.*

Table 1: TrajAuditor positive rates post unlearning (higher on $\mathcal{D}_r$, lower on $\mathcal{D}_f$ is better). The $\mathcal{D}_{f,1\%}$ and $\mathcal{D}_{f,5\%}$ denote the forget datasets with 1% and 5% of the original data to unlearn.

Task	Backbone	RETRAIN			FINE-TUNE			R-REWARD			TRAJDELETER			CURE		
		$\mathcal{D}_r$	$\mathcal{D}_{f,1\%}$	$\mathcal{D}_{f,5\%}$	$\mathcal{D}_r$	$\mathcal{D}_{f,1\%}$	$\mathcal{D}_{f,5\%}$	$\mathcal{D}_r$	$\mathcal{D}_{f,1\%}$	$\mathcal{D}_{f,5\%}$	$\mathcal{D}_r$	$\mathcal{D}_{f,1\%}$	$\mathcal{D}_{f,5\%}$	$\mathcal{D}_r$	$\mathcal{D}_{f,1\%}$	$\mathcal{D}_{f,5\%}$
Hopper	BEAR	85.9	22.7%	10.6%	85.7%	77.3%	77.0%	85.0%	90.9%	21.2%	84.4%	0%	0%	86.9%	0%	0%
	BCQ	82.7	0%	0%	82.0%	45.5%	43.4%	80.2%	0%	0%	80.7%	0%	0%	82.2%	0%	0%
	CQL	87.3	0%	0.2%	85.2%	13.6%	30.7%	82.7%	61.6%	28.7%	86.0%	0%	0.3%	87.7%	0%	0%
	IQL	85.7	18.0%	10.6%	85.0%	77.3%	81.4%	82.7%	68.2%	38.1%	83.0%	0%	0%	86.5%	0%	0%
	PLAS-P	21.1	0%	0%	56.9%	68.2%	82.3%	52.7%	50.0%	25.7%	66.0%	50.0%	26.3%	76.4%	23.6%	12.8%
	TD3+BC	84.6	9.1%	31.9%	84.3%	77.3%	72.6%	85.2%	86.4%	49.6%	85.1%	0%	0%	84.5%	0%	0%
	Average	**74.6%**	**8.3%**	**8.9%**	**79.8%**	**59.9%**	**64.6%**	**78.1%**	**59.5%**	**27.2%**	**80.9%**	**8.3%**	**4.4%**	**84.0%**	**3.93%**	**2.1%**
Half-Cheetah	BEAR	84.3%	20%	0%	84.7%	0%	3.0%	84.4%	30%	0%	85.0%	0%	0%	84%	0%	0%
	BCQ	74.1%	0%	0%	76.2%	30%	1%	78.5%	0%	0%	78.0%	0%	0%	82.9%	0%	0%
	CQL	80.1%	0%	0%	83.7%	0%	0%	86.1%	0%	0%	82.6%	0%	0%	88.1%	0%	0%
	IQL	76.8%	0%	0%	81.1%	10%	42%	82.7%	0%	0%	80.3%	0%	0%	85.1%	0%	0%
	PLAS-P	51.0%	0%	0%	70.0%	85%	22%	83.8%	65.0%	71.0%	80.9%	85%	68%	72.1%	32.5%	13.1%
	TD3+BC	72.8%	0%	0%	80.6%	70%	24%	79.1%	35.0%	0%	80.4%	0%	0%	85.6%	0%	0%
	Average	**73.2%**	**3.3%**	**0%**	**79.4%**	**32.5%**	**15.3%**	**82.4%**	**21.7%**	**11.8%**	**81.2%**	**14.2%**	**11.3%**	**82.9%**	**5.4%**	**3.4%**
Walker2D	BEAR	83.2%	8.7%	7.2%	83.8%	47.8%	66.9%	84.0%	27.5%	0.8%	84.3%	1.4%	0.3%	84.8%	0%	0%
	BCQ	80.7%	0%	0.6%	79.7%	36.2%	41.5%	79.6%	30.4%	0.3%	80.0%	0%	0%	80.6%	0%	0%
	CQL	85.7%	0%	0.5%	86.2%	13.0%	20.5%	84.9%	19.4%	6.3%	85.4%	0%	0.4%	86.8%	0%	0%
	IQL	78.3%	10.1%	8.1%	78.0%	55.1%	45.2%	77.8%	63.8%	0.9%	78.9%	0%	0%	78.7%	0%	0%
	PLAS-P	80.0%	11.6%	2.9%	82.5%	62.3%	74.4%	83.1%	72.5%	74.9%	81.9%	76.8%	73.2%	78.7%	12%	7.2%
	TD3+BC	84.6%	43.5%	4.9%	84.5%	81.2%	50.1%	85.3%	79.7%	4.9%	85.1%	0%	0.3%	85.0%	21%	10%
	Average	**82.1%**	**12.3%**	**4.0%**	**80.5%**	**47.1%**	**49.8%**	**82.5%**	**48.9%**	**14.7%**	**82.6%**	**13.0%**	**9.2%**	**82.4%**	**5.4%**	**3.4%**

Cor. 3 formalizes the key intuition of CURE: alignment-based gating selectively reduces coupling-induced retained drift while maintaining suppression strength on weakly aligned trajectories. The analysis above yields two testable predictions: *i)* lower-gated forget trajectories are associated with reduced retained-side critic drift in nearby retained regions, and *ii)* TD-based alignment correlates with the corresponding suppression-side interference signal. We empirically validate these in Sec. 5.3.

5 Experiments

Experiments are designed to answer the following questions: *(Q1)* Does CURE achieve a favorable trade-off among forgetting efficacy, retained utility, and efficiency under overlap? *(Q2)* Are the mechanism predictions in Sec. 4 supported empirically? *(Q3)* How robust is CURE to larger deletion ratios, λ, and repeated unlearning cycles?

5.1 Experiment Setup

Tasks & datasets. We evaluate on MuJoCo control environments [19] (including `Hopper`, `HalfCheetah`, and `Walker2D`) using D4RL `medium-expert` datasets [5], following [8]. 1% and 5% of data are randomly sampled to be unlearned.

Backbone offline RL algorithms. CURE is agent-agnostic, supporting unlearning for agents trained with diverse offline RL algorithms. We evaluate six common backbones [15]: bootstrapping error accumulation reduction (BEAR) [11], batch-constrained deep Q-learning (BCQ) [7], conservative Q-learning (CQL) [12], implicit Q-learning (IQL) [10], policy in the latent action space with perturbation (PLAS-P) [22], and twin delayed deep deterministic policy gradient plus behavioral cloning (TD3+BC) [6], all initialized from full-data pretrained models on $\mathcal{D}$.

Baselines. We compare CURE against four representative strategies. RETRAIN: We retrain the backbone on $\mathcal{D}_r$. FINE-TUNE: We continue training the pretrained model on $\mathcal{D}_r$ for the same number of updates as CURE, serving as a lightweight baseline that reflects simple continued learning after deletion. R-REWARD: We perturb the forget data by replacing rewards in $\mathcal{D}_f$ with uniform random values within the dataset reward range, and then fine-tune on the modified dataset. These baseline tests whether naively corrupting reward signals in $\mathcal{D}_f$ is sufficient to erase the corresponding behaviors. TRAJDELETER [8] suppresses critic values on $\mathcal{D}_f$ while encouraging convergence on $\mathcal{D}_r$, as a recent specialized baseline for RL unlearning.

Evaluation metrics. We evaluate along three dimensions: *Forgetting efficacy:* quantified by TRAJAUDITOR positive prediction rates [8] where lower rates imply lower detectable forget-supported influence. *Utility:* measured by post-unlearning return. *Efficiency:* measured by wall-clock runtime.

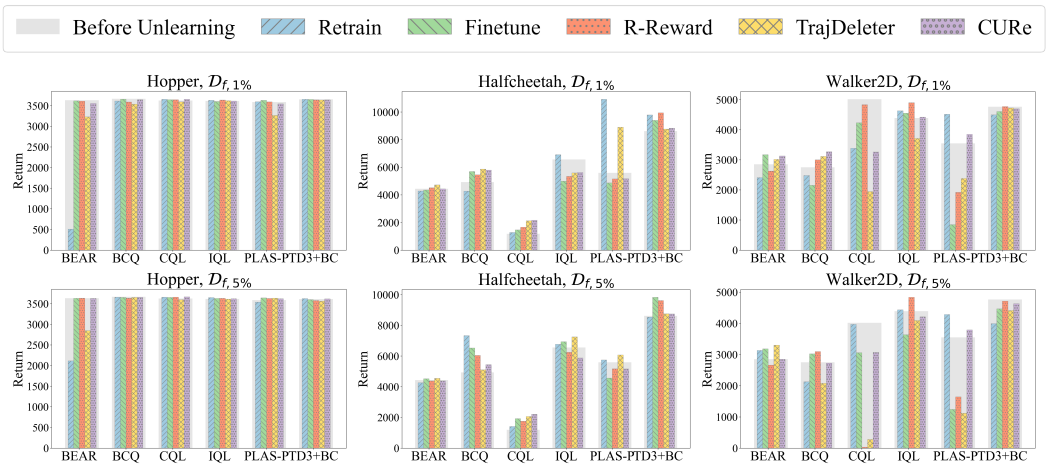
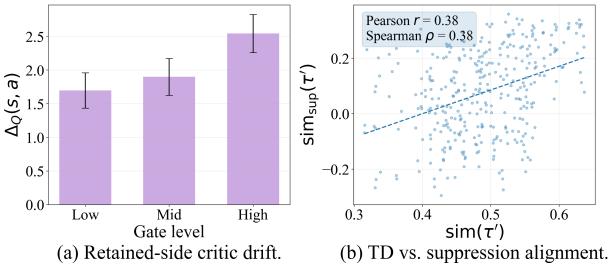


Figure 2: Post-unlearning returns across tasks and backbones with 1% and 5% unlearning.

Table 2: Unlearning time (minutes) averaged over backbones.

Methods	Tasks		
	Hopper	Half -Cheetah	Walker2D
RETRAIN	207.7	196.8	223.9
FINE-TUNE	2.1	2.1	2.1
R-REWARD	1.9	1.9	2.1
TRAJDELETER	2.6	2.9	3.4
CURE	**2.5**	**2.5**	**2.5**



(a) Retained-side critic drift. (b) TD vs. suppression alignment.

Figure 3: Mechanism validation. Lower-gates reduce drift; alignment correlates with interference.

5.2 Forgetting Efficacy and Utility under Overlap

Forgetting efficacy. Tab. 1 reports TRAJAUDITOR positive rates after unlearning. CURE consistently achieves the strongest forgetting efficacy under overlap, reducing $\mathcal{D}_f$ positives to near 0% across most tasks and backbones while keeping $\mathcal{D}_r$ positives high. This indicates lower auditor-detectable forget-supported influence with limited disruption to retained behavior. In contrast, FINE-TUNE largely preserves $\mathcal{D}_r$ but leaves substantial $\mathcal{D}_f$ positives, showing that continued training does not reliably erase detectable forget-supported influence. R-REWARD is unstable, and TRAJDELETER reduces $\mathcal{D}_f$ positives but remains non-zero on average; even RETRAIN can leave small residual positives. `Walker2D` is the hardest setting: CURE still lowers $\mathcal{D}_f$ positives while maintaining $\mathcal{D}_r$, though residual positives persist for PLAS-P and TD3+BC, consistent with the weaker forgetting observed for all baselines. Overall, CURE achieves the best forgetting efficacy, approaching low auditor detectability when $\mathcal{D}_f$ signatures are separable.

Unlearning utility. Fig. 2 reports post-unlearning returns over three tasks, six backbones, and two deletion ratios. Overall, CURE preserves utility: returns stay close to the pre-unlearning reference (gray bars) across most backbone-task pairs at both 1% and 5% unlearning, indicating no systematic performance loss from forgetting. This is most evident on `Hopper`, where CURE remains tightly aligned with the original performance for all backbones, while RETRAIN can underperform (*e.g.*, BEAR), suggesting that relearning from scratch with reduced data can be unstable or data-limited. On `HalfCheetah`, CURE remains competitive as deletion increases. In the harder `Walker2D` setting, baselines become markedly backbone-sensitive: FINE-TUNE, R-REWARD, and TRAJDELETER show higher variance and occasional collapses (notably TRAJDELETER on CQL at 5%). Overall, CURE provides robust utility preservation across tasks, backbones, and forgetting ratios.

Efficiency analysis. Tab. 2 shows that unlearning via CURE is orders of magnitude more efficient than RETRAIN across all three tasks and backbones. RETRAIN requires roughly 200-224 minutes per task, whereas FINE-TUNE, R-REWARD, and TRAJDELETER complete in only 2.6-3.4 minutes,

indicating that these methods avoid the dominant cost of training from scratch. Among the fast approaches, CURE's runtime is comparable, matching the speed of lightweight baselines while achieving substantially better forget-utility trade-offs. Overall, these numbers suggest CURE provides a practical path to deployment: it delivers unlearning at nearly the same runtime as simple fine-tuning, but at a tiny fraction of the retraining cost.

5.3 Mechanism Validation and Ablations

Empirical probing of the mechanism analysis. We validate two predictions from Sec. 4 on a representative setting (CQL on `Hopper`, 5% forgetting). *For the first prediction*, we measure retained-side critic drift before and after unlearning, $\Delta_Q(s,a) = \left|Q_{\phi'}(s,a) - Q_{\phi}(s,a)\right|$ on $(s,a) \in \mathcal{D}_r$, and group retained transitions by the gate level of their nearest forget transition (based on normalized Euclidean distance in observation space). As shown in Fig. 3(a), retained regions associated with lower-gated forget trajectories exhibit smaller critic drift, confirming that interference-based gating reduces unnecessary updates in retained-supported regions. *For the second prediction*, we test whether the TD-based alignment signal $\mathrm{sim}(\tau')$, serves as a meaningful proxy for suppression-side interference. Here, $\mathbf{g}_f(\tau')$ and $\mathbf{g}_r$ denote the TD gradients of the forget trajectory τ' and the retained dataset, respectively. We compare this signal with a suppression-direction proxy defined in the update direction: $\mathrm{sim}_{\mathrm{sup}}(\tau') = \cos\big(-\mathbf{g}_\Psi(\tau'), \mathbf{g}_r\big)$, where $\mathbf{g}_\Psi(\tau')$ is the gradient of the CURE objective, and $-\mathbf{g}_\Psi(\tau')$ reflects the effective update direction. Fig. 3(b) shows a moderate positive correlation between the two (Pearson $r = 0.38$, Spearman $\rho = 0.38$), indicating that TD-based alignment captures how unlearning updates interact with retained gradients. This supports the use of TD-gradient alignment as a lightweight proxy for suppression-side interference.

Ablations. We analyze the contribution of CURE 's key design choices, *i.e.*, interference-based gating and coupled actor suppression, on top of the base critic suppression mechanism. Tab. 3 compares CURE against variants that modify or remove these components. Removing the gate (NO-GATE) substantially worsens the forget-utility trade-off, leading to lower retained performance and return despite reduced forget influence. This aligns with Sec. 4, where uniform suppression amplifies coupling-driven drift by over-penalizing aligned trajectories. Removing actor-side suppression (NO-ACTOR) results in significantly higher residual forget detectability, showing that critic-only editing is insufficient to prevent forget behavior from resurfacing via policy updates.

Table 3: Ablations on `HalfCheetah` (5% forget).

Method	$\mathcal{D}_r$	$\mathcal{D}_f$	Return
CURE	88.1%	0%	4068.5
NO-GATE	76.1%	3.5%	3294.6
NO-ACTOR	87.3%	9.8%	4052.8
R-GATE	73.8%	8.2%	3326.6
HARD-GATE	82.3%	0%	3324.4
MAG-GATE	81.3%	0%	3284.7

We further examine variants that modify the gating mechanism: HARD-GATE, which binarizes alignment via a threshold 0; MAG-GATE, which replaces directional alignment with gradient magnitude-based weighting; and R-GATE, which assigns random trajectory weights. Both HARD-GATE (binary thresholding) and MAG-GATE (magnitude-only weighting) achieve near-zero $\mathcal{D}_f$, but incur substantial utility loss with lower returns (approximately 18%-19%) and degraded $\mathcal{D}_r$. This suggests that coarse or direction-agnostic suppression over-penalizes shared regions, degrading retained behavior. In contrast, R-GATE fails to reliably forget, demonstrating that arbitrary weighting does not capture the underlying interference structure. Overall, these results highlight that successful unlearning requires selective, interference-aware suppression rather than uniform or heuristic weighting. CURE is the only method that simultaneously achieves strong forgetting, high retained performance, and high return, demonstrating the importance of direction-aware, softly weighted gating for resolving overlap-induced interference.

5.4 Robustness and Sensitivity Analysis

Unlearning scalability. Tab. 4 evaluates CURE on `HalfCheetah` under deletion ratios 20%, 50%, averaged over backbones. CURE maintains a strong forgetting efficacy as deletions grow: at 20% it achieves low auditor-detectable forget influence while preserving retention, with a small return gap to TRAJDELETER. This utility advantage vanishes at 50%, where TRAJDELETER's forgetting degrades (high $\mathcal{D}_f$ positives), whereas CURE still drives $\mathcal{D}_f$ positives down with high $\mathcal{D}_r$ positives and stable returns. RETRAIN also deteriorates at 50% due to limited remaining data, leading to a marked return drop. Overall, CURE is both more data-efficient and more practical than retraining, while sustaining a favorable forget-utility trade-off at larger deletion ratios.

Table 4: Performance under increasing unlearning ratios on `HalfCheetah`.

Methods	Unlearning ratio 20%			Unlearning ratio 50%		
	$\mathcal{D}_r$	$\mathcal{D}_f$	Returns	$\mathcal{D}_r$	$\mathcal{D}_f$	Returns
RETRAIN	71.5%	0.1%	4871.9	35.2%	0%	4774.9
FINE-TUNE	83.9%	9.0%	5152.4	59.0%	3.3%	5058.9
R-REWARD	79.0%	12.1%	**6300.6**	72.6%	13.0%	**6390.2**
TRAJDELETER	75.3%	0.4%	5915.9	72.6%	11.9%	4909.7
CURE	**84.1%**	**0%**	5206.6	**87.6%**	**1.2%**	5105.9

Table 5: Performance under repeated unlearning cycles.

Times	Forget ratio	Tasks		
		`Hopper`	`Half-Cheetah`	`Walker2D`
1	1%	100.6%	102%	107.9%
	5%	100.6%	99.5%	109.1%
3	1%	99.7%	103%	109.6%
	5%	100.5%	100.1%	111.9%
7	1%	99.9%	102.3%	106.2%
	5%	100.5%	99.4%	106.6%
10	1%	100.2%	105.4%	109.6%
	5%	113.3%	100%	110.3%

Performance change with λ. Fig. 4 analyzes CURE 's sensitivity to λ on `HalfCheetah` with 5% unlearning. Across $\lambda \in \{0.1, 2, 5\}$, forgetting is invariant: TRAJAUDITOR positives on $\mathcal{D}_f$ remain at 0%, indicating consistent removal of $\mathcal{D}_f$-supported traces. Retention is stable as $\mathcal{D}_r$ positives change only mildly, suggesting CURE largely preserves the retained behavior distribution over a wide range of λ. In contrast, returns exhibit a clearer dependence on λ: moderate λ provides the best balance, maintaining strong returns while keeping high forgetting efficacy, whereas overly large λ reduces return, consistent with over-regularization of the update. This aligns with our theoretical analysis: λ scales the correction term that mediates the stability-plasticity trade-off, and thus mainly influences post-unlearning utility rather than the ability to eliminate $\mathcal{D}_f$ signatures.

Repeated unlearning. We evaluate CURE under repeated unlearn-relearn cycles ({1,3,7,10} cycles) on the same forget data with CQL [12], and report returns normalized by the original model in Tab. 5. Across tasks, performance remains stable with no degradation trend as the number of cycles increases, indicating that CURE does not accumulate harmful drift over repeated application. Moreover, `Walker2D` consistently stays above the original baseline, implying that repeated unlearning/relearning does not impair policy performance, likely due to regularization-like effects from the repeated parameter adjustments. The only notable outlier is `Hopper` at 10 cycles with a 5% ratio, which further supports the absence of compounding damage; instead, performance variations appear task- and ratio-dependent rather than systematically worsening. Overall, these findings suggest CURE is robust to repeated learning-unlearning.

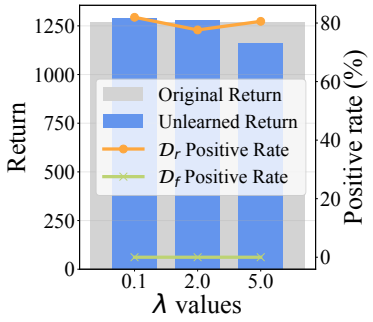


Figure 4: Sensitivity to λ.

6 Conclusion

Offline RL agents trained from static datasets increasingly require post-hoc deletion of specific data for privacy, safety, and governance. We propose CURE, a conservative unlearning framework that addresses retain-forget interference under state-action overlap by estimating per-trajectory influence via TD-gradient cosine alignment and selectively suppressing forget-supported critic values with coupled actor forgetting. By anchoring suppression to a retained reference, CURE performs minimum-necessary forgetting while controlling drift on retained behavior. Experiments on MuJoCo across six offline RL backbones show that CURE achieves the strongest auditor-level forgetting with near-original returns, at orders-of-magnitude lower cost than retraining.

Limitations. Gradient-based influence can be noisy, adds overhead, and introduces tunable hyperparameters. We evaluate on MUJOCO with TRAJAUDITOR, while practical deployments may require targeted deletions, stronger auditors, and richer observations. CURE is not a certified privacy mechanism; future work includes scalable influence estimation, broader offline RL settings, stronger auditing, and continual unlearning with stability guarantees.

Impacts. This work enables efficient unlearning in offline RL, supporting privacy and governance for data-driven systems (*e.g.*, "right-to-be-forgotten") and safer deployment by allowing post hoc removal of undesirable behaviors without retraining. Potential risks include misuse for accountability evasion, unintended degradation of model performance or safety, and fairness concerns arising

from biased selection of forget and retain sets. We mitigate these risks by explicitly evaluating both forgetting efficacy and retained utility, and by positioning unlearning as a complement to data auditing, documentation, and access controls. Overall, we expect these techniques to support more responsible lifecycle management of offline RL systems, while emphasizing the importance of careful dataset governance and transparent reporting.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback, which helped improve the presentation of this paper. Bijal Bharadva, Manuel Kelly, and Xin Zhang were supported in part by NSF awards IIS-2449864 and IIS-2617313. Yingxue Zhang was supported in part by the NSF under Grants IIS-2539629 (CAREER) and IIS-2617314.

References

[1] Silas Alberti, Kenan Hasanaliyev, Manav Shah, and Stefano Ermon. Data unlearning in diffusion models. *arXiv preprint arXiv:2503.01034*, 2025.

[2] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[3] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[4] Xing Fang, Qichao Zhang, Yinfeng Gao, and Dongbin Zhao. Offline reinforcement learning for autonomous driving with real world driving data. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3417–3422. IEEE, 2022.

[5] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

[6] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[7] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[8] Chen Gong, Kecen Li, Jin Yao, and Tianhao Wang. Trajdeleter: Enabling trajectory forgetting in offline reinforcement learning agents. *arXiv preprint arXiv:2404.12530*, 2024.

[9] Manuel Kelly, Yingxue Zhang, Fangzhou Lin, Yanhua Li, and Xin Zhang. Unlearning diffusion policies via relative fisher forgetting. In *Advances in Neural Information Processing Systems*, 2026.

[10] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *CoRR*, abs/2110.06169, 2021.

[11] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *CoRR*, abs/1906.00949, 2019.

[12] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[13] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[14] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[15] Takuma Seno and Michita Imai. d3rlpy: An offline deep reinforcement learning library. *Journal of Machine Learning Research*, 23(315):1–20, 2022.

[16] Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *Conference on Robot Learning*, pages 907–917. PMLR, 2022.

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

[18] Shengpu Tang, Maggie Makar, Michael Sjoding, Finale Doshi-Velez, and Jenna Wiens. Leveraging factored action spaces for efficient offline reinforcement learning in healthcare. *Advances in Neural Information Processing Systems*, 35:34272–34286, 2022.

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

[20] Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasediff: Erasing data influence in diffusion models. *arXiv preprint arXiv:2401.05779*, 2024.

[21] Dayong Ye, Tianqing Zhu, Congcong Zhu, Derui Wang, Kun Gao, Zewei Shi, Sheng Shen, Wanlei Zhou, and Minhui Xue. Reinforcement unlearning. *arXiv preprint arXiv:2312.15910*, 2023.

[22] Wenxuan Zhou, Sujay Bajracharya, and David Held. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, pages 1719–1735. PMLR, 2021.

Appendix

This appendix contains: *i)* proofs of propositions and corollaries in Sec. 4; *ii)* CURE algorithm; *iii)* additional implementation details; *iv)* additional experiment setup details, and *v)* supplementary experimental results.

A Proofs

A.1 Useful Lemmas

Lemma 4 (Descent lemma [14, 3])**.** *If f is L-smooth, then for any x, y,*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2.$$

Lemma 5 (γ-contraction [17])**.** *For any policy π, the Bellman operator T^π is a γ-contraction in $\|\cdot\|_\infty$:*

$$\|T^\pi V - T^\pi V'\|_\infty \leq \gamma \|V - V'\|_\infty.$$

A.2 Proof of Proposition 1

Proof. Apply Lemma 4 to $\Psi(\phi)$ with $x = \phi$ and $y = \phi^+$. Using $\phi^+ - \phi = -\alpha(\nabla_\phi \mathcal{L}_Q(\mathcal{D}_r; \phi) + \lambda \nabla_\phi \Psi(\phi))$ and expanding the inner products yields the stated bound. □

A.3 Proof of Proposition 2

Proof. Apply Lemma 4 to $\mathcal{L}_Q(\mathcal{D}_r; \phi)$ and substitute the same CURE update. Expanding gives the result. □

A.4 Proof of Corollary 3

Proof. By definition, $w(\tau')$ in Eq. (1) is non-increasing in $\mathrm{sim}(\tau')$. Using the decomposition in Eq. (4), down-weighting TD-aligned trajectories reduces their contribution to the expectation, and hence reduces the effective coupling term for fixed λ. □

B CURE Algorithm

At each iteration, CURE (Alg. 1) estimates gates $w(\tau')$ from TD-gradient alignment and performs one coupled critic-actor update using eq. (2)-eq. (3). We compute $\mathbf{g}_f(\tau')$ per trajectory or per grouped mini-trajectory batch to control overhead.

Algorithm 1 CURE Algorithm

Input: Original critic and actor Q_ϕ, π_θ; retained data $\mathcal{D}_r$; forget data $\mathcal{D}_f$; hyperparams λ, η; iterations K.
for $k = 1$ **to** K **do**
  Sample minibatches $\mathcal{B}_r \subset \mathcal{D}_r$ and $\mathcal{B}_f \subset \mathcal{D}_f$.
  Compute retained reference gradient $\mathbf{g}_r \leftarrow \nabla_\phi \frac{1}{|\mathcal{B}_r|} \sum_{\tau \in \mathcal{B}_r} \ell_{TD}(\tau; \phi)$.
  for each forget trajectory $\tau' \in \mathcal{B}_f$ **do**
    Compute forget gradient $\mathbf{g}_f(\tau') \leftarrow \nabla_\phi \ell_{TD}(\tau'; \phi)$.
    Compute cosine similarity $\mathrm{sim}(\tau') \leftarrow \frac{\langle \mathbf{g}_f(\tau'), \mathbf{g}_r \rangle}{\|\mathbf{g}_f(\tau')\| \|\mathbf{g}_r\|}$.
    Set forget weight $w(\tau')$ with eq. (1).
  end for
  Estimate $V_r(s)$ for states in $\mathcal{B}_f$ using $\mathcal{B}_r$.
  Update critic by minimizing eq. (2).
  Update actor by minimizing eq. (3).
end for
Output: Unlearned critic and actor $Q_{\phi'}, \pi_{\theta'}$;

Table 6: MuJoCo tasks and D4RL datasets used in our experiments.

Task	Dataset (D4RL)	Obs. Dim	Act. Dim	Act. Type	Data Size
`Hopper`	`hopper-medium-expert`	11	3	Cont.	2×10^6
`HalfCheetah`	`halfcheetah-medium-expert`	17	6	Cont.	2×10^6
`Walker2D`	`walker2d-medium-expert`	17	6	Cont.	2×10^6

C Additional Implementation Details

Estimating the retained baseline $V_r(s)$. We define $V_r(s) = \mathbb{E}_{a\sim\beta_r(\cdot|s)}[Q_\phi(s,a)]$, where $\beta_r(\cdot \mid s)$ denotes the retained-data action distribution at state s. Since exact state matches are uncommon in continuous-control offline RL, we approximate this quantity locally from the current retained minibatch $\mathcal{B}_r$. For each forget-state s, we find its k nearest retained states in $\mathcal{B}_r$ using Euclidean distance in normalized state space, and estimate $\hat{V}_r(s) = \frac{1}{k}\sum_{(s_i^r,a_i^r)\in\mathcal{N}_k(s;\mathcal{B}_r)} Q_\phi(s,a_i^r)$. That is, we reuse retained actions from states near s, but evaluate them at the query state s with the current critic. Unless otherwise stated, we use $k = 5$, normalize states dimension-wise using dataset statistics, and treat $\hat{V}_r(s)$ as fixed within the current update step. This approximation is sufficient in our setting because $V_r(s)$ is used only as a local retained reference in the conservative suppression term, rather than as a separate regression target.

D Additional Experiment Setup Details

We evaluate CURE on three MuJoCo continuous-control tasks-`Hopper`, `HalfCheetah`, and `Walker2D` [19]. Each environment provides a low-dimensional observation vector summarizing the robot's configuration and dynamics (*e.g.*, joint positions/velocities and body orientations). The action space is continuous. All offline datasets are from D4RL [5]. D4RL provides multiple data qualities (*e.g.*, `random`, `medium`, `medium-replay`, `medium-expert`); unless stated otherwise, we use the highest-performing variant per task, which is `medium-expert` for all three environments in our experiments (Table 6).

Backbone algorithms. We instantiate CURE as a plug-in wrapper over multiple offline RL backbones (*e.g.*, IQL, CQL, TD3+BC, BEAR, BCQ). Unless noted otherwise, we use the default hyperparameters provided by the corresponding open-source implementations and follow standard training/evaluation protocols for D4RL.

Backbone-specific implementation of actor suppression. Eq. (3) gives the unified actor-side forgetting objective used in our analysis. Its role is to discourage reproduction of forget actions at forget states, modulated by the same trajectory-level gate used in critic suppression. In implementation, this objective is instantiated according to the policy parameterization of each offline RL backbone, while keeping the underlying forgetting principle unchanged.

We write the actor objective as $\mathcal{L}_\pi^{\text{CURe}} = \mathcal{L}_\pi^{\text{base}}(\mathcal{D}_r;\theta) + \eta\,\mathbb{E}_{\tau'\sim\mathcal{D}_f}[w(\tau')\,\ell_{\text{sup}}(\tau')]$, where $\mathcal{L}_\pi^{\text{base}}(\mathcal{D}_r)$ is the backbone's original actor loss evaluated on retained data only, and ℓ_{sup} is instantiated according to the policy parameterization of the underlying backbone.

For backbones with an explicit stochastic actor, we instantiate $\ell_{\text{sup}} = \frac{1}{T}\sum_{t=1}^{T}\log\pi_\theta(a_t \mid s_t)$. For deterministic actors, we instantiate suppression in action space by discouraging the actor from matching the forget action, *e.g.*, $\ell_{\text{sup}}(\tau') = -\frac{1}{T}\sum_t \|\pi_\theta(s_t) - a_t\|_2^2$. For generative or latent-action policies, we apply the suppression term to the corresponding actor-related action-generation module so as to reduce regeneration of forget actions at forget states.

In all cases, the retained-data actor objective is left unchanged; CURE adds only a gated forget-side discouragement term during unlearning. Thus, Eq. (3) should be interpreted as a unified functional template rather than a literally identical parameterization across all backbones.

Table 7: Unlearning efficacy-utility on `HalfCheetah` with different data qualities averaged over backbones. Metrics: TRAJAUDITOR positive rates on $\mathcal{D}_r$ and $\mathcal{D}_f$, and returns.

Methods	`medium`			`medium-expert`			`expert`		
	$\mathcal{D}_m$	$\mathcal{D}_f$	Returns	$\mathcal{D}_m$	$\mathcal{D}_f$	Returns	$\mathcal{D}_m$	$\mathcal{D}_f$	Returns
RETRAIN	83.7%	23%	4740.4	73.2%	0%	4871.9	82.54%	15%	**11314.6**
FINE-TUNE	87.45%	62%	4749.7	79.4%	15.3%	5152.4	82.61%	47%	10272.1
R-REWARD	85.1%	17%	4755.6	82.4%	11.8%	**6300.6**	82.67%	15%	10629.4
TRAJDELETER	84.3%	14%	4699.9	81.2%	11.3%	5915.9	83.4%	14%	10545.5
CURE	**89%**	**0%**	**4776.9**	**82.8%**	**0%**	5206.6	**85.5%**	**0%**	10681.6

CURe hyperparameters. CURe introduces two main hyperparameters: the critic forgetting weight λ and the actor forgetting weight η (for actor-critic backbones). We also use the smooth hinge parameter β only for analysis; the implementation uses the hinge $[\cdot]_+$.

Across all tasks, we use the following defaults unless stated otherwise: $\lambda = 0.1, \quad \eta = 0.1$. We run CURe for $K = 150$ gradient steps. Gates $w(\tau')$ are recomputed once per unlearning iteration from a retained minibatch reference gradient and then treated as constants for the subsequent critic/actor update.

Evaluation. Returns are computed using 1000 evaluation episodes. Auditor metrics are computed using TRAJAUDITOR following TRAJDELETER [8].

E Additional Experiment Results

We provide additional experiment results to evaluate CURE performance given varied data quality.

Data quality. We study robustness to training data quality using D4RL `medium`, `expert`, and `medium-expert` datasets [5], deleting 5% of data. Tab. 7 reports TRAJAUDITOR positives on $\mathcal{D}_r$/$\mathcal{D}_f$ and returns on `HalfCheetah`. CURE is the only method that achieves complete forgetting (0% $\mathcal{D}_f$ positives) while maintaining strong retention across all data regimes. Baselines leave residual $\mathcal{D}_f$ detectability, especially FINE-TUNE under lower-quality data, while R-REWARD and TRAJDELETER reduce but do not eliminate it. In terms of utility, CURE attains best or near-best returns under `medium` and `expert`, and remains competitive under `medium-expert`, where RETRAIN can achieve higher return but still leaves a sizable $\mathcal{D}_f$ footprint. Overall, CURE delivers the most consistent forgetting-retention trade-off across data qualities.

Repeated unlearning. We evaluate CURE under repeated unlearn-relearn cycles ({1,3,7,10} cycles) on the same forget data with the CQL [12] backbone, and report returns normalized by the original model in Tab. 8. The results show no degradation trend as the number of cycles increases: across all tasks, the normalized returns remain stable and typically fluctuate within a small band around the original performance for `Hopper` and `HalfCheetah` under most settings. This indicates that CURE 's updates do not accumulate harmful drift over repeated application, suggesting stable retention of task competence even when unlearning is invoked multiple times. Moreover, `Walker2D` consistently stays above the original baseline, implying that repeated unlearning/relearning does not impair policy performance, likely due to regularization-like effects from the repeated parameter adjustments. The only notable outlier is `Hopper` at 10 cycles with a 5% ratio, which further supports the absence of compounding damage; instead, performance variations appear task- and ratio-dependent rather than systematically worsening. Overall, these findings suggest CURE is robust to repeated learning-unlearning operations, maintaining utility rather than suffering from cumulative forgetting-induced instability.

Table 8: Return with percentage of improvement over the original return under repeated unlearning cycles on $\mathcal{D}_f$ with the CQL [12] backbone.

Times	Unlearning ratio	Tasks		
		Hopper	HalfCheetah	Walker2D
1	1%	3630.4 (**+0.6%**)	4170.0 (**+2%**)	3955.2 (**+7.9%**)
	5%	3633.0 (**+0.6%**)	4068.5 (**-0.5%**)	4000.1 (**+9.1%**)
3	1%	3600.1 (**-0.3%**)	4210.8 (**+3%**)	4018.2 (**+9.6%**)
	5%	3629.0 (**+0.5%**)	4094.8 (**+0.1%**)	4102.7 (**+11.9%**)
7	1%	3608.1 (**-0.1%**)	4184.9 (**+2.3%**)	3892.3 (**+6.2%**)
	5%	3628.2 (**+0.5%**)	4064.6 (**-0.6%**)	3905.2 (**+6.6%**)
10	1%	3617.3 (**+0.2%**)	4311.0 (**+5.4%**)	4015.9 (**+9.6%**)
	5%	4089.5 (**+13.3%**)	4088.5 (**+0%**)	4041.9 (**+10.3%**)

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction clearly state the main claims: CURE addresses offline RL unlearning under state-action overlap via TD-gradient interference scoring and soft-gated actor-critic suppression, with theoretical guarantees (Sec. 4) and empirical validation (Sec. 5) supporting the forget-utility trade-off claims across MuJoCo benchmarks.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Limitations are explicitly discussed in the Conclusion (Sec. 6): gradient-alignment estimates can be noisy and add computational overhead; evaluation is restricted to MuJoCo with TrajAuditor; CURE is not a certified privacy mechanism; and hyperparameter sensitivity is noted.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: All propositions and corollaries in Sec. 4 are stated with explicit assumptions. Proof sketches appear in the main paper; full proofs are provided in Appx. A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The algorithm is fully described in Alg. 1, hyperparameters are specified in Appx. D, backbone implementations follow open-source defaults, and an anonymized codebase is provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

Justification: Code is released anonymously. All datasets used (D4RL) are publicly available. Instructions for reproducing experiments with backbone-specific implementation details are provided in Appx. D.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Sec. 5 and Appx. D specify tasks (Hopper, HalfCheetah, Walker2D), datasets (D4RL medium-expert), backbone algorithms, baselines, forget ratios, evaluation protocol, and all key hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: Due to the high computational cost of running six backbone algorithms across three tasks and multiple deletion ratios, error bars are not reported. Results are averaged over backbones in ablation and scalability tables (Tab. 3, 4), which partially mitigates variance concerns.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Wall-clock runtimes are reported in Tab. 2 to compare efficiency across methods. All experiments are run on 6000 Ada.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: This work does not involve human subjects, sensitive data collection, or deployment of high-risk systems. The research conforms with the NeurIPS Code of Ethics; potential societal risks are discussed in the Impacts section of Sec. 6.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The Impacts paragraph in Sec. 6 discusses positive impacts (privacy/governance compliance, safer deployment) and negative risks (accountability evasion, unintended performance degradation, fairness concerns from biased forget/retain set selection), along with mitigation strategies.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: This paper proposes an algorithm for offline RL unlearning and does not release pretrained language models, image generators, or scraped datasets that pose high misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: All external assets are properly cited, and all backbone algorithm papers. D4RL and MuJoCo are publicly available for research use under their respective open licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: An anonymized code repository implementing CURE as a plug-in wrapper over multiple offline RL backbones is released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: This paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on standard simulated MuJoCo control benchmarks with publicly available offline datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: No human subjects are involved in this research. IRB approval is not required.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    [No].

    Justification: Large language models are not part of the proposed methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.