TRACER: Trust-Calibrated Offline-to-Online Reinforcement Learning

Yutong Zhang [1] **Yaoran Yang** [1]

Abstract

Offline-to-online reinforcement learning (O2O RL) can reuse historical data, but stale or corrupted logs can also anchor online learning to the wrong model. We study a local alternative to fixed replay ratios: each time-state-action region receives a trust weight determined by offline coverage, online-offline reward/transition disagreement, and a confidence radius. The resulting algorithm, TRACER, uses this weight to mix empirical models, modulate optimism, and apply a local suspicion penalty. In finite-horizon tabular MDPs, we prove a high-probability interpolation bound with complete in-text proofs: clean covered regions contribute low-variance offline error, while identifiable shifted regions contribute an exponentially attenuated offline-bias term. Empirically, we evaluate 17 tabular algorithms and ablations on 18 controlled O2O tasks. These experiments should be read as a controlled mechanism study, not as a D4RL/MuJoCo or exact published-code comparison: ROAD-, ARB-, WSRL-, and RLPD-style baselines are tabular proxies. Within this scope, TRACER obtains the highest aggregate final return (82.86$\pm$3.19), best 10th-percentile return (65.1), and lowest failure rate (2.8%). Regime-level results are mixed–TRACER is positive in 6 of 15 family/corruption cells and negative in all Layered cells against the strongest cell-wise proxy–which clarifies both the promise and current limits of local trust calibration.

1. Introduction

O2O RL is attractive when historical interaction logs are available but online exploration is expensive or risky. Offline data can stabilize early learning and supply sparse-reward coverage; the same data can be harmful when rewards were hacked, transitions came from a stale simulator, actions were logged through faulty actuators, or behavior coverage is narrow. The practical question is therefore not whether prior data are useful in the aggregate, but *which decision regions remain reliable after deployment begins*.

A fixed offline replay ratio is a coarse answer to a local reliability problem. Consider an MDP with a safe bridge and a shortcut. If the shortcut reward is corrupted in the log, retaining all offline data biases the agent toward the shortcut; discarding all offline data loses useful bridge coverage; annealing a global ratio can still retain corrupted transitions for too long or erase clean transitions too early. The correct decision is spatially heterogeneous: trust the bridge, distrust the shortcut, and revisit ambiguous regions online. This is the operating regime targeted by TRACER.

Prior O2O methods instantiate different stability-plasticity compromises. RLPD mixes offline and online replay with stabilized off-policy learning (Ball et al., 2023); Cal-QL and OPT address value-function calibration before online fine-tuning (Nakamoto et al., 2023; Shin et al., 2025); WSRL studies when offline replay can be dropped (Zhou et al., 2024); ROAD and ARB adapt replay at global or behavior-aware levels (Liu et al., 2025; Song et al., 2025); and robust O2O work studies corrupted logs and heavy-tailed Q bias (He et al., 2025; Guo et al., 2026). These mechanisms are valuable, but they usually expose a dataset-level or critic-level knob rather than a local reliability certificate.

We propose TRACER, which assigns a trust weight $\tau_t(h, s, a) \in [0, 1]$ to every time-state-action tuple. High trust preserves offline evidence; low trust shifts the planner toward online estimates. The weight is intentionally local: an agent can retain clean bridge transitions while rejecting a corrupted shortcut, or keep reliable reward data while relearning shifted dynamics elsewhere. The same scalar is used consistently in model mixing, exploration, and conservatism, which makes the mechanism auditable rather than a collection of independent heuristics.

The paper makes three claims, stated with explicit scope.

- **Algorithmic principle.** A local trust map can unify coverage gating, online-offline discrepancy detection, calibrated optimism, and conservative penalties.

[1]School of Mathematics, Sichuan University, Chengdu, 610065, China. Correspondence to: Yutong Zhang <yutongzhang@stu.scu.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

- **Tabular theory.** In finite-horizon tabular MDPs, local trust yields a value interpolation bound that separates clean coverage benefits from identifiable offline bias. All mathematical claims are proved in the main text.
- **Controlled empirical evidence.** On a self-contained tabular suite, TRACER has the best aggregate and lower-tail performance, but it is not uniformly dominant per regime. We do not claim standard continuous-control benchmark superiority.

2. Related Work

Offline-to-online RL. RLPD, Cal-QL, WSRL, and OPT study how to initialize or fine-tune online RL with offline data (Ball et al., 2023; Nakamoto et al., 2023; Zhou et al., 2024; Shin et al., 2025). Their central concern is sample-efficient online improvement without catastrophic forgetting or value overestimation. ROAD and ARB make replay more adaptive (Liu et al., 2025; Song et al., 2025). TRACER is closest in motivation to adaptive replay, but its control variable is a tuple-level trust weight rather than a single replay ratio or behavior score. This distinction matters when the same dataset contains both high-value clean regions and high-value corrupted regions.

Offline RL and conservative learning. Offline RL methods such as CQL, IQL, TD3+BC, and BCQ address extrapolation error without online correction (Kumar et al., 2020; Kostrikov et al., 2021; Fujimoto and Gu, 2021; Fujimoto et al., 2019). Their pessimism is primarily a safeguard against unsupported actions. In O2O RL the agent can actively resolve uncertainty, so pessimism should be conditional: clean supported regions should be exploited, unsupported regions should be explored, and contradicted regions should be penalized. TRACER separates these cases through coverage, online residuals, and confidence.

Robustness, uncertainty, and hybrid theory. RPEX and LAROO motivate robustness to corrupted or heavy-tailed offline sources (He et al., 2025; Guo et al., 2026). Hybrid RL theory shows that offline and online samples can complement one another under coverage assumptions (Li et al., 2023; Tan et al., 2024). We add a bound whose terms depend explicitly on local trust, coverage, online evidence, and corruption magnitude. The result is not a neural-function-approximation theorem; it is a finite-horizon tabular certificate explaining why a local trust map can interpolate between offline and online estimation.

3. Problem Setup

We consider a finite-horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, P, r, \rho)$ with rewards in $[-1, 1]$. The learner receives an offline dataset $\mathcal{D}_0$ of tuples (h, s, a, r, s') generated by a source process $\mathcal{M}_0 = (\mathcal{S}, \mathcal{A}, H, P_0, r_0, \rho)$ that may differ from deployment. It then collects online data $\mathcal{D}_t$ in the true MDP for T episodes. For $x = (h, s, a)$, let $n_0(x), n_t(x)$ be offline and online counts, and let $\widehat{r}_0(x)$, $\widehat{P}_0(\cdot|x)$ and $\widehat{r}_t(x)$, $\widehat{P}_t(\cdot|x)$ be empirical reward and transition estimates. A fixed replay method uses a constant offline weight. We instead learn $\tau_t(x)$ and interpolate locally.

The theory uses the value-span constant $L_H = 2H$. This constant is conservative but convenient: because one-step rewards lie in $[-1, 1]$, any remaining-horizon value function has span at most $2H$, so transition total-variation error contributes at most $L_H \,\mathrm{TV}(\cdot, \cdot)$ to value error.

Definition 1 (Local source bias)**.** *Let $x = (h, s, a)$ and $L_H = 2H$. The local offline bias at x is*

$$B_0(x) = |r_0(x) - r(x)| + L_H \,\mathrm{TV}(P_0(\cdot|x), P(\cdot|x)). \quad (1)$$

A tuple is clean when $B_0(x) = 0$, shifted when $B_0(x) > 0$, and identifiable when online evidence can make the source-deployment discrepancy statistically visible.

This formulation deliberately permits heterogeneous datasets. The source MDP can be correct on most tuples while corrupted on a small subset. The learner is not told which tuples are shifted. Its decision at time t is to choose a policy using both $\mathcal{D}_0$ and $\mathcal{D}_t$ while avoiding irreversible commitment to biased source evidence. We evaluate expected deployment return, not source return.

4. Method: Trust-Calibrated Replay and Planning

4.1. Design criteria

A useful O2O trust statistic should satisfy four criteria. First, it should be *coverage aware*: no amount of nominal optimism should make a method trust absent offline actions. Second, it should be *falsifiable*: once online samples contradict the source, the method should be able to reduce offline influence locally. Third, it should be *calibrated*: a few early online samples should not erase a large reliable log. Fourth, it should be *actionable*: the statistic should affect both estimation and data collection. TRACER implements these criteria with a single scalar $\tau_t(x)$.

4.2. Trust score and component roles

TRACER computes an online-offline discrepancy

$$\Delta_t(x) = |\widehat{r}_0(x) - \widehat{r}_t(x)| + \lambda_P \,\mathrm{TV}\Big(\widehat{P}_0(\cdot|x), \widehat{P}_t(\cdot|x)\Big), \quad (2)$$

setting $\Delta_t(x) = 0$ before online evidence is observed. The transition coefficient λ_P controls the reward-transition scale. The confidence radius is

$$b_t(x) = c\left(\frac{1}{\sqrt{n_0(x)+1}} + \frac{1}{\sqrt{n_t(x)+1}}\right), \quad (3)$$

Table 1. Operational role of each TRACER component. The terms are not interchangeable: coverage prevents extrapolation, residuals identify mismatch, confidence prevents premature rejection, and the bonus/penalty pair changes behavior.

Component	Statistical role	Failure mode addressed
$g_0(x)$	Requires offline support before reusing logs	Prevents unsupported offline extrapolation
$\Delta_t(x)$	Tests local online-offline compatibility	Detects reward hacks, stale dynamics, action faults
$b_t(x)$	Delays rejection under scarce online evidence	Avoids overreacting to early noisy samples
$B_t(x)$	Directs exploration toward low-trust regions	Forces validation where source data are suspicious
$\Lambda_t(x)$	Discounts contradicted logged evidence	Reduces anchoring to corrupted but well-covered tuples

and the offline coverage factor is $g_0(x) = n_0(x)/(n_0(x) + \kappa)$. The trust weight is

$$\tau_t(x) = g_0(x)\,\sigma\left(\frac{b_t(x) - \widetilde{\Delta}_t(x)}{\eta}\right), \qquad (4)$$
$$\sigma(z) = \frac{1}{1+e^{-z}}.$$

Here $\widetilde{\Delta}_t$ is a clipped or smoothed residual in the implementation; clipping only prevents single-sample numerical spikes and is not used to hide persistent mismatch. Coverage prevents unsupported extrapolation, the residual tests compatibility, the confidence term avoids rejecting offline data from a few online samples, and η controls how sharply trust changes. In the controlled experiments the default tabular parameters are $\lambda_P = 0.10$, $c = 2.05$, $\eta = 0.12$, $\kappa = 4$, exploration coefficient $\beta = 0.12$, and suspicion coefficient $\lambda_\Lambda = 0.08$.

The score is not intended as a Bayesian posterior probability that the source is clean. It is a calibrated gating statistic. This distinction is important: a posterior would require a generative model over corruptions, whereas TRACER only needs a monotone mapping that preserves offline influence when residuals are within the confidence radius and attenuates it when residuals exceed that radius.

4.3. Trusted model and value update

The trusted empirical model is

$$\widehat{P}_t^\tau(\cdot|x) = \tau_t(x)\widehat{P}_0(\cdot|x) + (1-\tau_t(x))\widehat{P}_t(\cdot|x), \qquad (5)$$
$$\widehat{r}_t^\tau(x) = \tau_t(x)\widehat{r}_0(x) + (1-\tau_t(x))\widehat{r}_t(x). \qquad (6)$$

Algorithm 1 TRACER: Trust-Calibrated Offline-to-Online RL

Input: offline data $\mathcal{D}_0$, budget T, constants $c, \eta, \kappa, \lambda_P, \beta, \lambda_\Lambda$
Estimate $(\widehat{r}_0, \widehat{P}_0, n_0)$ from $\mathcal{D}_0$; initialize online counts $n_t = 0$
for $t = 1, \dots, T$ **do**
  Estimate online $(\widehat{r}_t, \widehat{P}_t, n_t)$ from collected data
  Compute Δ_t, b_t, τ_t by Eqs. (2)–(4)
  Form $\widehat{r}_t^\tau, \widehat{P}_t^\tau$, bonus B_t, and penalty Λ_t
  Plan by backward DP; execute an ϵ-greedy episode; append it to $\mathcal{D}_t$
end for

Backward dynamic programming uses

$$Q_h(s,a) = \widehat{r}_t^\tau(x) + B_t(x) - \Lambda_t(x) + \widehat{P}_t^\tau(\cdot|x)^\top V_{h+1},$$
$$V_h(s) = \max_a Q_h(s,a), \qquad (7)$$

where $B_t(x) = \beta\sqrt{1-\tau_t(x)}/\sqrt{n_t(x)+1}$ encourages online validation where trust is low, and $\Lambda_t(x) = \lambda_\Lambda(1 - \tau_t(x))\mathbb{I}\{n_0(x) > 0\}\widetilde{\Delta}_t(x)$ penalizes offline-supported regions that disagree with online observations.

The optimism and suspicion terms have different roles. B_t is largest when online evidence is scarce and trust is low, pushing the policy to validate uncertain regions. Λ_t is active when a tuple has offline support but the online residual is high; it prevents the planner from repeatedly exploiting a high-reward but contradicted source artifact. Removing either term changes the operating point from local trust calibration to a simpler replay interpolation.

Complexity. In a tabular implementation, computing trust is $O(HSA)$ plus the transition support cost, and planning is the usual finite-horizon dynamic program. The memory overhead is one additional scalar map for τ_t and one residual map. This overhead is small relative to storing empirical transition counts. In a neural implementation, the analogous cost would come from estimating calibrated uncertainty and residuals in representation space; we leave that extension to future work rather than claiming it here.

5. Theory and Complete Proofs

We prove the theoretical claims directly in the main text. All statements are finite-horizon and tabular. For a policy π, let d_h^π denote the occupancy distribution over (s,a) at time h in the deployment MDP $\mathcal{M}$. For compactness write $x = (h,s,a)$ and define the one-step scaled model error

$$\mathcal{E}(M, M'; x) = |r_M(x) - r_{M'}(x)| + L_H\,\mathrm{TV}(P_M(\cdot|x), P_{M'}(\cdot|x)). \qquad (8)$$

Assumption 1 (Identifiable local mismatch)**.** *For any shifted*

tuple x, once $n_t(x) > 0$, the expected discrepancy satisfies

$$\mathbb{E}[\Delta_t(x)] \geq |r_0(x) - r(x)| + \lambda_P \mathrm{TV}(P_0(\cdot|x), P(\cdot|x)) - \xi_t(x), \quad (9)$$

where $\xi_t(x)$ is a statistical tolerance that decreases with $n_0(x)$ and $n_t(x)$. For clean tuples, $B_0(x) = 0$.

The assumption does not require every corrupted tuple to be visited immediately. It only states that, conditional on visiting a tuple, the observed residual is informative about the local source-deployment gap. Non-identifiable corruption remains a fundamental limitation: if online exploration never visits a shifted tuple and no structural generalization links it to visited tuples, no residual-based method can certify that shift.

Proposition 1 (Trust separation)**.** *Fix x and let $m \geq 0$. If $\widetilde{\Delta}_t(x) \leq b_t(x) - m$, then*

$$\tau_t(x) \geq \frac{g_0(x)}{1 + e^{-m/\eta}} \geq \frac{g_0(x)}{2}. \quad (10)$$

If $\widetilde{\Delta}_t(x) \geq b_t(x) + m$, then

$$\tau_t(x) \leq g_0(x) e^{-m/\eta}. \quad (11)$$

Proof. By Eq. (4),

$$\tau_t(x) = g_0(x)\sigma(z_t(x)), \qquad z_t(x) = \frac{b_t(x) - \widetilde{\Delta}_t(x)}{\eta}. \quad (12)$$

If $\widetilde{\Delta}_t(x) \leq b_t(x) - m$, then $z_t(x) \geq m/\eta$. Since the sigmoid is monotone increasing,

$$\sigma(z_t(x)) \geq \sigma(m/\eta) = \frac{1}{1 + e^{-m/\eta}}. \quad (13)$$

Because $m \geq 0$, $e^{-m/\eta} \leq 1$, and hence $(1 + e^{-m/\eta})^{-1} \geq 1/2$. Multiplying by $g_0(x)$ proves the first claim. If $\widetilde{\Delta}_t(x) \geq b_t(x) + m$, then $z_t(x) \leq -m/\eta$. Again using monotonicity,

$$\sigma(z_t(x)) \leq \sigma(-m/\eta) = \frac{1}{1 + e^{m/\eta}}. \quad (14)$$

For $m \geq 0$, $1 + e^{m/\eta} \geq e^{m/\eta}$, so $(1 + e^{m/\eta})^{-1} \leq e^{-m/\eta}$. Multiplication by $g_0(x)$ proves the second claim. □

Lemma 1 (Uniform empirical model concentration)**.** *Let $N = H|\mathcal{S}||\mathcal{A}|(T+1)$ and let $\delta \in (0,1)$. For $i \in \{0, t\}$ define, for every tuple x, the reward and transition radii*

$$\varepsilon_i^r(x) = \begin{cases} 2, & n_i(x) = 0, \\ \sqrt{\dfrac{2\log(8N/\delta)}{n_i(x)}}, & n_i(x) > 0, \end{cases} \quad (15)$$

$$\varepsilon_i^P(x) = \begin{cases} 1, & n_i(x) = 0, \\ \sqrt{\dfrac{\log(8\max\{1, 2^{|\mathcal{S}|} - 2\}N/\delta)}{2n_i(x)}}, & n_i(x) > 0. \end{cases} \quad (16)$$

Set $\varepsilon_i(x) = \varepsilon_i^r(x) + L_H \varepsilon_i^P(x)$. With probability at least $1 - \delta$, simultaneously for every $t \leq T$ and every $x = (h, s, a)$,

$$|\widehat{r}_0(x) - r_0(x)| \leq \varepsilon_0^r(x), \quad \mathrm{TV}\Big(\widehat{P}_0(\cdot|x), P_0(\cdot|x)\Big) \leq \varepsilon_0^P(x), \quad (17)$$

$$|\widehat{r}_t(x) - r(x)| \leq \varepsilon_t^r(x), \quad \mathrm{TV}\Big(\widehat{P}_t(\cdot|x), P(\cdot|x)\Big) \leq \varepsilon_t^P(x). \quad (18)$$

Consequently, on the same event,

$$\mathcal{E}(\mathcal{M}_0, \widehat{\mathcal{M}}_0; x) \leq \varepsilon_0(x), \quad (19)$$

$$\mathcal{E}(\mathcal{M}, \widehat{\mathcal{M}}_t; x) \leq \varepsilon_t(x). \quad (20)$$

Proof. We prove the reward and transition parts separately and then take a union bound. If $n_i(x) = 0$, the reward bound is trivial because both true and empirical rewards lie in $[-1, 1]$, so their absolute difference is at most 2. The transition bound is also trivial because total variation distance between probability measures is at most 1.

Assume first that $n_i(x) = n > 0$ is fixed. Conditional on the event that tuple x is sampled n times, the observed rewards are conditionally independent bounded random variables with conditional mean $r_i(x)$, where $i = 0$ refers to the source MDP and $i = t$ refers to the deployment MDP. For offline data this is ordinary i.i.d. sampling from the source logging process. For online data, the times at which x is visited are stopping times, but the Markov property implies that the reward noise and next-state draw following the k-th visit to x have the same conditional law as a fresh draw from the deployment kernel at x. Thus the standard bounded-difference concentration argument applies to the subsequence of visits to x.

Hoeffding's inequality for variables in an interval of length 2 gives

$$\Pr(|\widehat{r}_i(x) - r_i(x)| > u \mid n_i(x) = n) \leq 2\exp\left(-\frac{nu^2}{2}\right). \quad (21)$$

Choosing $u = \sqrt{2\log(8N/\delta)/n}$ makes this probability at most $\delta/(4N)$. Since there are at most N relevant tuple-time-count events after unioning over $H|\mathcal{S}||\mathcal{A}|$ tuples and $T + 1$ online sample sizes, the probability that any reward concentration statement fails is at most $\delta/4$ for the source and at most $\delta/4$ for the online process.

For transitions, apply the Weissman multinomial concentration inequality to the empirical distribution over $\mathcal{S}$:

$$\Pr\Big(\|\widehat{P}_i(\cdot|x) - P_i(\cdot|x)\|_1 > v \mid n_i(x) = n\Big) \leq (2^{|\mathcal{S}|} - 2)\exp\left(-\frac{nv^2}{2}\right), \quad (22)$$

with the factor replaced by 1 when $|\mathcal{S}| = 1$. Because $\mathrm{TV}(p, q) = \frac{1}{2}\|p - q\|_1$, the event $\mathrm{TV}(\widehat{P}_i, P_i) > u$ implies

$\|\widehat{P}_i - P_i\|_1 > 2u$. Therefore

$$\Pr\Big(\mathrm{TV}(\widehat{P}_i(\cdot|x), P_i(\cdot|x)) > u \mid n_i(x) = n\Big) \le \max\{1, 2^{|\mathcal{S}|} - 2\} \exp(-2nu^2). \tag{23}$$

Choosing

$$u = \left(\frac{\log(8\max\{1, 2^{|\mathcal{S}|} - 2\}N/\delta)}{2n}\right)^{1/2} \tag{24}$$

makes the failure probability at most $\delta/(8N)$ for each empirical transition statement. A union bound over source and online estimates, all tuples, and all online sample sizes gives probability at least $1 - \delta$ that Eqs. (17)–(18) hold simultaneously. Multiplying the transition inequalities by L_H and adding the reward inequalities gives the two displayed bounds on $\mathcal{E}$. □

Lemma 2 (Finite-horizon simulation bound)**.** *Let M and M' be two finite-horizon MDPs on the same state and action spaces with rewards in $[-1, 1]$. For any policy π,*

$$\left|V^\pi_{M,1}(\rho) - V^\pi_{M',1}(\rho)\right| \le \sum_{h=1}^{H} \mathbb{E}_{d^\pi_{M,h}}[\mathcal{E}(M, M'; h, s, a)]\,. \tag{25}$$

where $d^\pi_{M,h}$ is the occupancy under M.

Proof. Let $V^\pi_{M,h}$ and $V^\pi_{M',h}$ be the time-h value functions under the same policy π in M and M', with terminal values $V^\pi_{M,H+1} = V^\pi_{M',H+1} = 0$. Define

$$D_h(s) = |V^\pi_{M,h}(s) - V^\pi_{M',h}(s)|. \tag{26}$$

For a fixed state s, write $P_M = P_M(\cdot|x)$ and $P_{M'} = P_{M'}(\cdot|x)$. Bellman expansion and Jensen's inequality give

$$D_h(s) \le \mathbb{E}_{a\sim\pi_h(\cdot|s)}\Big[|r_M(x) - r_{M'}(x)| + \left|P_M^\top V^\pi_{M,h+1} - P_{M'}^\top V^\pi_{M',h+1}\right|\Big]. \tag{27}$$

Add and subtract $P_{M'}^\top V^\pi_{M,h+1}$ inside the second absolute value. Then

$$\begin{aligned}\left|P_M^\top V^\pi_{M,h+1} - P_{M'}^\top V^\pi_{M',h+1}\right| &\le \left|(P_M - P_{M'})^\top V^\pi_{M,h+1}\right| \\ &\quad + P_{M'}^\top |V^\pi_{M,h+1} - V^\pi_{M',h+1}|.\end{aligned} \tag{28}$$

The first term is controlled by total variation and span. Since rewards lie in $[-1, 1]$, every remaining-horizon value lies in $[-H, H]$, hence $\mathrm{span}(V^\pi_{M,h+1}) \le 2H = L_H$. For any bounded function f, $|(p - q)^\top f| \le \mathrm{span}(f)\mathrm{TV}(p, q)$, so

$$\left|(P_M - P_{M'})^\top V^\pi_{M,h+1}\right| \le L_H\, \mathrm{TV}(P_M(\cdot|x), P_{M'}(\cdot|x))\,. \tag{29}$$

Substituting this inequality into the previous display yields

$$D_h(s) \le \mathbb{E}_{a\sim\pi_h(\cdot|s)}\left[\mathcal{E}(M, M'; x) + P_{M'}^\top D_{h+1}\right]. \tag{30}$$

Iterating this recursion gives a sum of one-step model errors plus a terminal term. The terminal term is zero because $D_{H+1} = 0$. Writing the resulting expectation along trajectories generated by M gives Eq. (25). Using M' in the recursive propagation would produce the analogous bound under M' occupancy; the displayed form is sufficient for our theorem because the performance target is the deployment MDP. □

Theorem 1 (High-probability trust interpolation)**.** *Let $\widehat{\mathcal{M}}^\tau_t$ be the empirical MDP with reward $\widehat{r}^\tau_t$ and transition $\widehat{P}^\tau_t$. Under the concentration event of Lemma 1, for every policy π and every episode $t \le T$,*

$$\left|V^\pi_{\mathcal{M}} - V^\pi_{\widehat{\mathcal{M}}^\tau_t}\right| \le \sum_{h=1}^{H} \mathbb{E}_{(s,a)\sim d^\pi_h}\Big[\tau_t(x)(B_0(x) + \varepsilon_0(x)) + (1 - \tau_t(x))\varepsilon_t(x)\Big]. \tag{31}$$

If a shifted tuple has measured residual margin $\widetilde{\Delta}_t(x) - b_t(x) \ge m \ge 0$, then its offline-bias contribution inside Eq. (31) *is at most $g_0(x)e^{-m/\eta}B_0(x)$.*

Proof. Fix a tuple $x = (h, s, a)$. The trusted reward estimate is the convex combination

$$\widehat{r}^\tau_t(x) = \tau_t(x)\widehat{r}_0(x) + (1 - \tau_t(x))\widehat{r}_t(x). \tag{32}$$

Therefore, by the triangle inequality and nonnegativity of $\tau_t(x)$ and $1 - \tau_t(x)$,

$$\begin{aligned}|r(x) - \widehat{r}^\tau_t(x)| &= \left|\tau_t(x)(r - \widehat{r}_0) + (1 - \tau_t(x))(r - \widehat{r}_t)\right| \\ &\le \tau_t(x)|r - \widehat{r}_0| + (1 - \tau_t(x))|r - \widehat{r}_t|.\end{aligned} \tag{33}$$

For the offline reward term, add and subtract the source mean reward $r_0(x)$:

$$|r(x) - \widehat{r}_0(x)| \le |r(x) - r_0(x)| + |r_0(x) - \widehat{r}_0(x)|. \tag{34}$$

On the event of Lemma 1, the second term is at most $\varepsilon^r_0(x)$. The first term is the reward component of $B_0(x)$. Similarly, $|r(x) - \widehat{r}_t(x)| \le \varepsilon^r_t(x)$ for the online empirical reward.

For transitions, convexity of total variation gives

$$\begin{aligned}\mathrm{TV}(P, \widehat{P}^\tau_t) &= \mathrm{TV}\Big(P, \tau_t\widehat{P}_0 + (1 - \tau_t)\widehat{P}_t\Big) \\ &\le \tau_t(x)\mathrm{TV}(P, \widehat{P}_0) + (1 - \tau_t(x))\mathrm{TV}(P, \widehat{P}_t).\end{aligned} \tag{35}$$

The offline transition term is controlled by the triangle inequality for total variation:

$$\mathrm{TV}(P, \widehat{P}_0) \le \mathrm{TV}(P, P_0) + \mathrm{TV}(P_0, \widehat{P}_0). \tag{36}$$

On the concentration event, $\mathrm{TV}(P_0, \widehat{P}_0) \le \varepsilon^P_0(x)$, while $\mathrm{TV}(P, \widehat{P}_t) \le \varepsilon^P_t(x)$. Multiplying Eq. (35) by L_H, adding Eq. (33), and substituting the definitions of B_0, ε_0, and ε_t gives

$$\begin{aligned}\mathcal{E}(\mathcal{M}, \widehat{\mathcal{M}}^\tau_t; x) &\le \tau_t(x)(B_0(x) + \varepsilon_0(x)) \\ &\quad + (1 - \tau_t(x))\varepsilon_t(x).\end{aligned} \tag{37}$$

Finally, apply the finite-horizon simulation bound in Lemma 2 with $M = \mathcal{M}$ and $M' = \widehat{\mathcal{M}}^\tau_t$, and substitute Eq. (37). This gives Eq. (31).

For the margin statement, if $\widetilde{\Delta}_t(x) - b_t(x) \ge m$, Proposition 1 gives $\tau_t(x) \le g_0(x)e^{-m/\eta}$. The offline-bias part of Eq. (31) at tuple x is $\tau_t(x)B_0(x)$, hence it is bounded by $g_0(x)e^{-m/\eta}B_0(x)$. □

The theorem explains why local trust can outperform a global replay ratio: it can keep the low-variance offline term on clean covered tuples without retaining the full offline-bias term on shifted tuples. It also explains when TRACER may not help. If offline data are uniformly clean and high coverage, then aggressive distrust can be worse than a strong replay baseline; if the online policy never visits shifted tuples, residuals cannot reduce trust. These cases appear empirically in the Layered family and motivate the regime-level analysis below.

Corollary 1 (Comparison to a fixed replay weight)**.** *Let a fixed method use offline weight $\alpha \in [0,1]$ at every tuple, forming the empirical model $\widehat{\mathcal{M}}_t^{\alpha}$ by replacing $\tau_t(x)$ with α. On the concentration event of Lemma 1, the local-trust upper bound in Theorem 1 is no larger than the fixed-weight upper bound whenever*

$$\sum_{h=1}^{H} \mathbb{E}_{(s,a)\sim d_h^{\pi}}[(\tau_t(x)-\alpha)(B_0(x)+\varepsilon_0(x)-\varepsilon_t(x))] \le 0. \tag{38}$$

Thus local trust is most valuable when tuples with high source bias receive smaller weights than tuples where offline empirical error is lower than online empirical error.

Proof. Theorem 1 gives the local-trust upper bound

$$U_\tau = \sum_{h=1}^{H} \mathbb{E}_{d_h^{\pi}}[\tau_t(x)(B_0(x)+\varepsilon_0(x)) + (1-\tau_t(x))\varepsilon_t(x)]. \tag{39}$$

Applying the same theorem to the fixed interpolation model, with $\tau_t(x)$ replaced everywhere by α, gives

$$U_\alpha = \sum_{h=1}^{H} \mathbb{E}_{d_h^{\pi}}[\alpha(B_0(x)+\varepsilon_0(x)) + (1-\alpha)\varepsilon_t(x)]. \tag{40}$$

Subtracting the two displays and collecting terms gives

$$\begin{aligned} U_\tau - U_\alpha &= \sum_{h=1}^{H} \mathbb{E}_{d_h^{\pi}}[(\tau_t(x)-\alpha)(B_0(x)+\varepsilon_0(x)) \\ &\qquad -(\tau_t(x)-\alpha)\varepsilon_t(x)] \\ &= \sum_{h=1}^{H} \mathbb{E}_{d_h^{\pi}}[(\tau_t(x)-\alpha) \\ &\qquad \cdot(B_0(x)+\varepsilon_0(x)-\varepsilon_t(x))]. \end{aligned} \tag{41}$$

Therefore condition (38) is exactly the condition $U_\tau \le U_\alpha$. The qualitative statement follows because $B_0(x)+\varepsilon_0(x)-\varepsilon_t(x)$ is large and positive precisely where source bias or source estimation error dominates online error; assigning $\tau_t(x) < \alpha$ on those tuples reduces the bound, while assigning $\tau_t(x) > \alpha$ can help where offline evidence is more accurate than online evidence. □

6. Experiments

6.1. Scope, benchmark, and baselines

The execution environment used for this study did not include Gymnasium, Minari, D4RL, or MuJoCo. We therefore report a fully controlled finite-horizon suite designed to isolate local trust behavior. The suite contains Bridge, Grid, and Layered MDP families with clean, reward-corrupted, dynamics-shifted, mixed, action-corrupted, and low-coverage settings. Offline data are generated in a source MDP by a soft-optimal behavior policy; online interaction occurs in the deployment MDP.

The three families stress different aspects of O2O learning. Bridge tasks contain narrow high-value paths where wrong rewards can create tempting shortcuts. Grid tasks distribute reward and transition errors across spatially local regions, making local mismatch detection important. Layered tasks have strong logged structure and comparatively smooth online corrections; they test whether the method can avoid unnecessary suspicion when offline data remain useful. The corruption types are also local: reward corruption changes observed source rewards, dynamics corruption changes transition kernels, mixed corruption changes both, and action corruption perturbs the action actually executed by the source process. Low-coverage clean tasks isolate the effect of $g_0(x)$.

We compare 17 algorithms/variants over 18 tasks, 4 seeds, and 90 online episodes, yielding 110,160 policy-evaluation records. Values are exact expected deployment returns, not rollout estimates. Baselines named ROAD-style, ARB-style, WSRL-style, and FixedMix-50/RLPD are deliberately labeled as *tabular proxies*; they test the corresponding design ideas inside our suite but are not exact reproductions of published neural implementations. This naming is conservative and avoids overstating the strength of the empirical comparison.

The controlled suite uses a common tabular planner and changes only the mechanism for retaining offline evidence. **OfflineOnly** plans exclusively with the source empirical model and therefore measures the cost of source bias. **Online-UCB** ignores the source and measures the cost of relearning from deployment samples. **FixedMix-25/50/75** use constant offline interpolation weights and isolate the weakness of a global replay knob. **DecayReplay** begins with a high offline weight and anneals it over online episodes. **Pessimistic** applies a uniform conservative penalty to source-supported actions. **OptimisticPretrain** uses source estimates for initialization but emphasizes online optimism thereafter. **RobustTrim** removes high-residual tuples after online evidence accumulates. The ROAD-, ARB-, WSRL-, and RLPD-style entries implement the corresponding replay ideas in this tabular protocol, but they should not be cited as

Table 2. Controlled-suite results across 72 task-seed runs. P10 is the 10th percentile of final return; Fail is the fraction of runs with final return below 50. Higher return/AUC/P10 and lower Fail/Cost are better.

Method	Final return	AUC	P10	Fail	Cost
TRACER	82.86 ± 3.19	70.54 ± 3.83	65.1	2.8%	1.171
ARB-style	77.69 ± 4.16	64.98 ± 4.19	54.2	6.9%	1.161
RobustTrim	76.74 ± 4.03	65.84 ± 4.11	49.9	11.1%	1.322
FixedMix-50/RLPD	75.36 ± 4.65	64.30 ± 4.77	49.4	11.1%	1.178
DecayReplay	74.88 ± 4.51	65.52 ± 4.61	46.1	16.7%	1.361
ROAD-style	74.64 ± 4.77	64.52 ± 4.76	49.8	11.1%	1.325
OptimisticPretrain	73.38 ± 3.43	60.95 ± 3.79	51.2	8.3%	1.268
FixedMix-25	71.56 ± 4.71	60.52 ± 4.68	44.7	19.4%	1.204
WSRL-style	69.54 ± 3.92	53.57 ± 3.90	45.9	16.7%	1.184
Online-UCB	68.75 ± 3.86	58.30 ± 3.93	47.5	19.4%	1.002
FixedMix-75	64.37 ± 7.20	55.64 ± 7.56	23.2	27.8%	1.691
Pessimistic	53.47 ± 10.34	50.70 ± 10.35	-12.2	33.3%	1.725
OfflineOnly	26.61 ± 12.58	26.37 ± 12.37	-43.8	55.6%	3.129

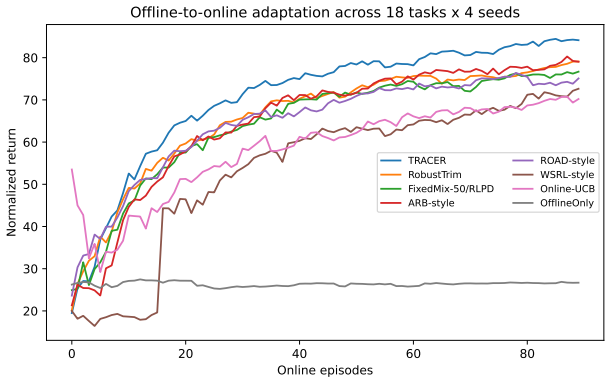

Figure 1. Average adaptation curves over all controlled tasks. TRACER improves early performance over Online-UCB and avoids the worst failures of fixed offline retention.

results for the original neural algorithms.

This distinction is central to the empirical claim. The comparison answers a mechanism question: under identical tabular dynamics, does local trust calibration improve aggregate and lower-tail behavior relative to fixed, annealed, robust-trimmed, and proxy adaptive replay rules? It does not answer whether TRACER outperforms published code on MuJoCo, D4RL, Minari, or other large-scale continuous-control suites. We therefore report both aggregate gains and cell-wise losses.

6.2. Aggregate results and lower-tail robustness

Table 2 shows that TRACER has the strongest aggregate return and lower-tail profile in this controlled suite. Its mean final return is 82.86±3.19, compared with 77.69±4.16 for ARB-style and 75.36±4.65 for FixedMix-50/RLPD. The more diagnostic result is lower-tail robustness: TRACER has the best P10 return (65.1) and lowest failure rate (2.8%), suggesting that the aggregate gain comes largely from avoiding severe failures rather than winning every setting.

Figure 1 shows the average learning profile. Online-UCB avoids source bias but pays an exploration cost; OfflineOnly and high fixed replay can perform well when the source is clean but fail badly under shift; adaptive proxy methods

Table 3. Paired tests for TRACER versus tabular proxy baselines. Intervals are bootstrap intervals over task-seed paired differences.

Baseline	Mean gain	95% bootstrap CI	paired p
OfflineOnly	56.25	[44.08, 68.74]	8.74e-13
Pessimistic	29.39	[19.07, 39.99]	1.41e-06
FixedMix-75	18.49	[11.88, 25.85]	1.41e-06
Online-UCB	14.11	[9.72, 17.96]	3.63e-09
WSRL-style	13.32	[9.07, 17.46]	2.63e-08
FixedMix-25	11.31	[7.09, 15.42]	1.57e-06
OptimisticPretrain	9.48	[6.43, 12.36]	2.49e-08
ROAD-style	8.22	[4.36, 12.21]	1.74e-04
DecayReplay	7.99	[3.84, 12.57]	7.39e-04
FixedMix-50/RLPD	7.50	[3.63, 11.57]	6.50e-04
RobustTrim	6.13	[2.64, 9.68]	1.30e-03
ARB-style	5.17	[1.69, 8.87]	6.62e-03

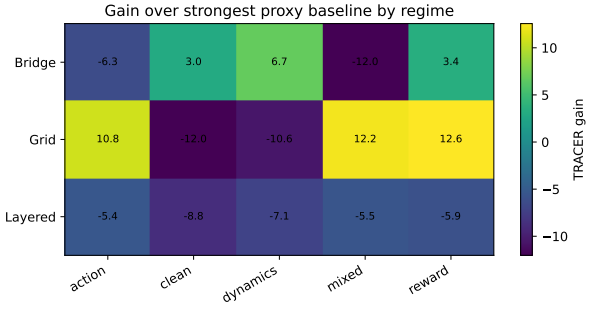

Figure 2. Gain over the strongest tabular proxy baseline in each family/corruption cell. TRACER is positive in 6/15 cells and negative in all Layered cells, so the result should not be read as uniform regime dominance.

reduce that failure mode. TRACER's curve is less about a large asymptotic jump and more about reducing early and mid-training exposure to corrupted source artifacts while still exploiting clean coverage.

The paired comparisons in Table 3 provide a stricter view than the aggregate ranking. They separate consistent paired improvements from improvements driven by a subset of hard tasks. This is important for O2O evaluation because a method can look strong in mean score while losing to simpler replay in clean or highly structured regimes.

6.3. Regime-level results are mixed

Figure 2 addresses a key failure mode of aggregate reporting. TRACER is strongest in several localized-corruption settings, especially Grid reward/action/mixed and Bridge dynamics/reward, but it loses all Layered cells to the best cell-wise proxy. The Layered family has high-quality local structure where conservative suspicion can underuse useful logged transitions. This is a limitation of the current tabular instantiation, not a contradiction of the aggregate result.

Tables 4 and 5 reinforce the same point. TRACER improves the Bridge lower tail relative to fixed replay and improves Grid failures relative to Online-UCB, but the best fixed/proxy replay methods are stronger in Layered tasks. We therefore avoid the claim that TRACER is a uniformly best algorithm. The supported claim is narrower: local

Table 4. Regime-level gains over the strongest tabular proxy baseline.

Family	Corruption	TRACER	Best proxy	Gain
Bridge	action	73.40	79.69	-6.29
Bridge	clean	89.82	86.78	3.04
Bridge	dynamics	89.50	82.75	6.75
Bridge	mixed	75.56	87.58	-12.02
Bridge	reward	90.93	87.50	3.44
Grid	action	67.14	56.33	10.81
Grid	clean	71.94	83.95	-12.01
Grid	dynamics	73.47	84.02	-10.55
Grid	mixed	81.14	68.97	12.17
Grid	reward	83.86	71.29	12.57
Layered	action	85.77	91.13	-5.36
Layered	clean	87.83	96.64	-8.81
Layered	dynamics	89.84	96.94	-7.10
Layered	mixed	91.22	96.76	-5.55
Layered	reward	90.53	96.45	-5.93

Table 5. Family-wise robustness. Each cell reports mean/P10/failure-rate for final deployment return.

Method	Bridge	Grid	Layered
TRACER	84.8/80.4/8%	74.9/58.9/0%	88.8/81.6/0%
ARB-style	74.3/41.5/12%	64.7/52.9/8%	94.1/89.8/0%
ROAD-style	68.6/33.7/17%	60.9/48.9/17%	94.4/89.3/0%
FixedMix-50/RLPD	69.2/45.5/21%	61.7/47.7/12%	95.2/92.5/0%
Online-UCB	84.9/69.6/0%	49.2/40.5/58%	72.2/64.6/0%

trust gives a better aggregate risk profile when clean and corrupted source regions coexist.

6.4. Mechanism diagnostics, ablations, and sensitivity

Figure 3 should be interpreted narrowly. It does not by itself prove a dramatic "learning-to-trust" trajectory: the current hyperparameters make average trust change slowly. The diagnostic instead confirms that shifted tuples exhibit larger online-offline residuals and that the implementation operates conservatively. This supports the need for the sensitivity and larger-benchmark studies described below.

Ablations in Figure 4 and Table 7 show that reward discrepancy is essential in this suite, while transition discrepancy matters in dynamics-shifted tasks. The result should not be over-generalized: in continuous-control domains with smoother reward misspecification and richer dynamics error, the relative importance of reward and transition residuals may change.

Tables 8 and 9 describe two important stress tests. The coverage stress test isolates the $g_0(x)$ gate: when logged data are sparse, trust should saturate slowly rather than treating a few source samples as a reliable model. The sensitivity table shows that TRACER is not parameter-free. Overly high optimism ($\beta = 0.18$) is harmful on the representative slice, while moderate changes to confidence scale and temperature are less destructive. These observations match the theory: the bound improves only when trust assigns lower weights to biased regions without unnecessarily discarding clean low-variance source evidence.

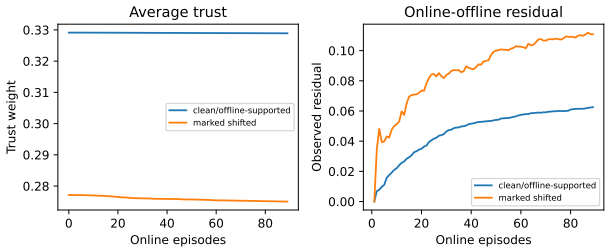

Figure 3. Trust diagnostic with corrected denominators: trust is averaged over offline-supported tuples, while residuals are averaged over online-visited tuples. The average trust gap is present early and changes slowly, indicating a conservative coverage-dominated operating point; residuals are nevertheless larger on marked shifted tuples.

Table 6. Trust/residual diagnostic averaged over episodes 75–89. "Marked" denotes the source-sensitive subset used by the diagnostic; in clean tasks it is a control subset rather than a genuinely corrupted set.

Corruption	$\bar{\tau}$ clean	$\bar{\tau}$ marked	Residual clean	Residual marked
clean	0.332	0.215	0.075	0.072
reward	0.326	0.295	0.035	0.117
dynamics	0.331	0.257	0.068	0.053
mixed	0.330	0.330	0.054	0.193
action	0.322	0.329	0.061	0.152

7. Discussion and Limitations

The main conceptual message is that O2O agents should expose reliability as a local object. A trust map is useful both for control and for diagnosis: high trust identifies reusable prior evidence, while low trust highlights regions requiring online validation. This view also changes how empirical results should be read. Aggregate return alone is insufficient; a local trust method should be evaluated on whether it reduces lower-tail failures, whether it identifies source-deployment residuals, and whether it avoids over-penalizing clean high-coverage data.

The present validation is intentionally controlled. It does not establish state-of-the-art performance on neural continuous-control benchmarks, and it does not replace exact comparisons with published RLPD, ROAD, ARB, or WSRL code. The theory is tabular and assumes identifiable local mismatch; neural extensions require calibrated uncertainty and representation stability assumptions. Finally, the average trust curves reveal that our default confidence setting is conservative. Future work should combine this local trust mechanism with standard Minari/D4RL-style adapters and learned uncertainty estimators.

Two failure modes should be checked first in any extension. The first is *excess suspicion*: when the offline source is clean and structured, a trust penalty can slow down exploitation relative to fixed replay. The second is *unobserved corruption*: if online exploration never visits the shifted tuple, residual-based trust cannot reject it. The current algorithm mitigates the second case through low-trust optimism, but this is not a formal reachability guarantee. Reporting

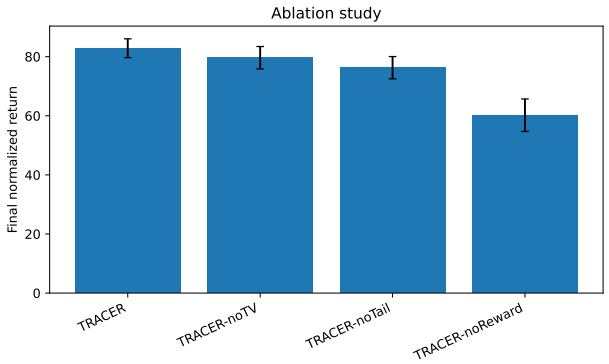

Figure 4. Ablations. Removing reward disagreement is most damaging; removing transition disagreement or calibrated trust strength also reduces performance.

Table 7. Ablation results.

Variant	Final return	AUC	Cost
TRACER	82.86 ± 3.19	70.54 ± 3.83	1.171
TRACER-noTV	79.64 ± 3.80	67.05 ± 3.98	1.102
TRACER-noTail	76.28 ± 3.77	65.33 ± 3.94	1.163
TRACER-noReward	60.19 ± 5.50	53.23 ± 3.25	1.153

both mean score and lower-tail score is therefore essential, because either failure mode can be hidden by averages.

8. Conclusion

We introduced TRACER, a trust-calibrated O2O RL framework that decides locally when to retain or abandon offline evidence. The method combines coverage, online-offline disagreement, confidence calibration, optimism, and a suspicion penalty. In tabular MDPs, a trust interpolation theorem separates the value of clean covered data from the cost of identifiable offline bias, and the proof follows from explicit concentration, simulation, and convexity arguments. Controlled experiments show strong aggregate and lower-tail robustness, while also exposing non-uniform regime performance and conservative trust dynamics. The result is best viewed as a principled step toward local reliability-aware O2O RL, not as a finished large-scale benchmark claim.

Table 8. Clean-task coverage stress test. Low coverage reduces the value of offline evidence and tests whether the coverage gate avoids over-trusting sparse logs.

Method	Medium coverage clean	Low coverage clean
TRACER	85.6 ± 3.3	80.8 ± 3.6
ARB-style	80.4 ± 4.4	84.2 ± 3.7
ROAD-style	79.5 ± 5.2	81.6 ± 4.4
FixedMix-50/RLPD	79.4 ± 5.1	82.5 ± 3.7
Online-UCB	68.7 ± 5.7	69.0 ± 4.9

Table 9. One-factor sensitivity on a representative slice: clean and reward-corrupted Bridge/Grid/Layered tasks, two seeds. This is not a full hyperparameter sweep.

Setting	Final return	AUC	Cost
default	85.51 ± 7.62	74.70 ± 7.25	0.891
conf=1.2	88.95 ± 4.68	79.09 ± 3.89	0.833
conf=0.8	88.56 ± 4.85	79.45 ± 5.83	0.927
temp=0.05	87.64 ± 7.46	77.27 ± 7.19	0.926
temp=0.20	89.66 ± 3.04	78.25 ± 5.07	0.867
kappa=8	86.92 ± 6.07	73.47 ± 7.69	0.839
beta=0.06	91.78 ± 3.75	80.38 ± 5.05	0.838
beta=0.18	76.67 ± 11.06	67.20 ± 7.97	1.054
penalty=0.04	89.69 ± 2.82	78.14 ± 5.28	0.799
penalty=0.16	92.35 ± 2.09	77.14 ± 5.41	0.685

References

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, 2023.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-QL: Calibrated offline RL pretraining for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, 2023.

Zhou, Z., Peng, A., Li, Q., Levine, S., and Kumar, A. Efficient online reinforcement learning fine-tuning need not retain offline data. arXiv:2412.07762, 2024.

Shin, Y., Kim, J., Jung, W., Hong, S., Yoon, D., Jang, Y., Kim, G.-H., Chae, J., Sung, Y., Lee, K., and Lim, W. Online pre-training for offline-to-online reinforcement learning. In *International Conference on Learning Representations*, 2025.

Liu, X., Yu, T., and Li, S. Adaptive offline data replay in offline-to-online reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, 2025.

Song, C., Lee, J., and Park, J. Adaptive replay buffer for offline-to-online reinforcement learning. arXiv:2512.10510, 2025.

He, L., Ye, D., Tan, J., Wang, X., and Shen, L. Robust policy expansion for offline-to-online RL under diverse data corruption. In *Advances in Neural Information Processing Systems*, 2025.

Guo, R., Yang, R., Liu, L., Shen, J., Wu, G., Wang, J., and Li, B. Tackling heavy-tailed Q-value bias in offline-to-online reinforcement learning with Laplace-robust modeling. In *International Conference on Learning Representations*, 2026.

Li, G., Zhan, W., Lee, J. D., Chi, Y., and Chen, Y. Reward-agnostic fine-tuning: Provable statistical benefits of hybrid reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Tan, K., Fan, W., and Wei, Y. Hybrid reinforcement learning breaks sample size barriers in linear MDPs. arXiv:2408.04526, 2024.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit Q-learning. arXiv:2110.06169, 2021.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2019.