CALIBRATE BEFORE YOU LEARN: COVERAGE-AWARE REWARDS FOR OFFLINE RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning has access to trajectory outcomes before training begins, but early temporal-difference updates still rely on a critic that has not learned to predict them. We propose coverage-aware reward calibration to use these outcomes as additional supervision. Before critic training, we fit a shared state potential to return-to-go labels on a nearest-neighbor graph, with local geometry controlling each label's influence. During training, we add the potential difference to the reward, amplify the original reward component, and gradually reduce both changes while retaining the base learner's update rules. Under the stated terminal convention, potential-based shaping preserves optimal action ordering at fixed coefficients. We relate potential error to behavior-advantage information and characterize when calibrated weights improve on uniform smoothing within locally constant-value regions. A two-step analytical example connects this improvement to the first Bellman update. Experiments cover sparse navigation, dexterous manipulation, and continuous control. Matched ablations show gains beyond reward scaling alone and support graph smoothing and geometry-based weighting, with additional controls across offline learners. These results support using precomputed outcomes to assist offline TD learning.

1 INTRODUCTION

Offline reinforcement learning (RL) learns from a fixed dataset without further environment interaction (Levine et al., 2020; Fu et al., 2020). The dataset already records what happened after each action: trajectory returns can be computed before training starts. Yet a one-step temporal-difference (TD) target uses only the immediate reward and a learned estimate of future return. When rewards are sparse or weak, that estimate can dominate early updates before the critic is accurate (Yue et al., 2023; Qiu et al., 2026; Kim et al., 2025). This raises a practical question: *can observed trajectory outcomes help train the critic before it learns to predict those outcomes itself?*

Offline RL methods usually address a different difficulty: policies may select poorly supported actions. ReBRAC, IDQL, and SCQ refine behavior regularization, implicit value learning, and conservative estimation, respectively (Tarasov et al., 2023; Hansen-Estruch et al., 2023; Shimizu et al., 2024). Our method complements these mechanisms by changing the reward in TD targets while retaining the learner's policy constraints and update rules.

Using returns in this way requires care. A return-to-go label describes one behavior continuation, not the best achievable outcome from a state. Nearby states may have different returns, and some regions have much less data support than others. Treating every label as equally reliable can give an isolated observation too much influence. We therefore need a shared state function that combines observed outcomes while accounting for the local data geometry.

We propose *coverage-aware reward calibration*. Before critic training, we fit return-to-go labels on a nearest-neighbor graph. The graph couples nearby states, while geometry-based weights control how strongly each label is followed. A well-supported label receives more weight. A less-supported label relies more on its graph neighbors. We bound the resulting state potential and add its discounted difference to the reward. A separate scale adjusts the strength of the original reward. Both changes decay during training, so later updates use rewards closer to those of the original task.

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

Potential-based reward shaping (PBRS) preserves optimal action ordering for any bounded potential under the stated terminal convention (Ng et al., 1999; Wiewiora, 2003). Within this policy-preserving class, the choice of potential determines which information is available in a one-step reward. We use behavior advantages as a reference: an accurate behavior value turns the expected shaped reward into an advantage signal (Schulman et al., 2016). Our analysis traces potential-estimation error through this identity and separates the graph fit's bias from its propagated label noise. In locally constant-value regions with uncorrelated label noise, weights sufficiently aligned with label precision improve on uniform smoothing. The comparison quantifies how much calibration error can be tolerated relative to the heterogeneity of the labels, with finite-smoothing and preprocessing terms treated explicitly.

An analytical two-step MDP in Appendix D.2 illustrates how calibrated weighting can reveal a delayed action gap in the first Bellman update. Geometry supplies a practical proxy for the label weights, evaluated through matched benchmark ablations of scaling, smoothing, and weighting. The full method has the highest reported score on all three ablation tasks, with additional controls across learners. A five-dataset return-alignment diagnostic examines the information retained by the potential. These results support using offline outcomes to complement TD learning while retaining the learner's treatment of unsupported actions.

2 Related Work

Offline constraints and reward scale. Policy constraints, conservative values, and model-based pessimism limit distribution-shift errors in offline RL (Kumar et al., 2020; Yu et al., 2020; Kidambi et al., 2020; Fujimoto & Gu, 2021). ReBRAC revisits TD3+BC through regularization and implementation choices (Tarasov et al., 2023). IDQL generalizes IQL with diffusion-based policy extraction (Hansen-Estruch et al., 2023). Strategically Conservative Q-Learning (SCQ) focuses CQL-style pessimism on actions far from the data (Shimizu et al., 2024). Cal-QL and WSRL address offline-to-online value calibration and warm starts (Nakamoto et al., 2023; Zhou et al., 2025). PARS combines reward scaling, layer normalization, and infeasible-action penalties (Kim et al., 2025). Our potential instead adds observed outcome information to TD rewards while retaining the learner's constraints. DOGE uses distance to data to constrain policy improvement (Li et al., 2023). Our geometry weights control return-label influence before critic training.

Potential shaping and outcome supervision. Potential-based reward shaping (PBRS) preserves optimal action ordering under the appropriate discounted terminal convention (Ng et al., 1999). Its links to Q-value initialization and dynamic potentials explain how intermediate rewards can change without changing the underlying action preferences (Wiewiora, 2003; Devlin & Kudenko, 2012). GAE interprets value-based shaping through advantage estimation (Schulman et al., 2016). Building on this relation, we analyze how graph fitting and label weighting affect the advantage information supplied by an offline potential. Offline shaping methods differ in the source of that potential: causal state-value upper bounds address confounded data, while STO-RL uses LLM-generated temporal subgoals for sparse-reward tasks (Li et al., 2025; Gu et al., 2026). Other uses of outcomes change policy supervision: Decision Transformer and RvS condition policies on returns or goals, and PREFORL contrasts successful and failed trajectories (Chen et al., 2021; Emmons et al., 2022; Qiu et al., 2026). Our construction instead combines observed returns and local geometry in a bounded state potential. Keeping it fixed during critic training separates the offline outcome estimate from the learned critic, while its annealed coefficient controls how strongly it enters the reward.

Critic initialization and TD geometry. Observed returns can also help before TD updates begin. Jelley et al. (2024) pretrain the actor and critic with supervised objectives, including Monte Carlo value regression for the critic. Our use of returns does not initialize the critic: it supplies a separate potential whose differences enter later TD targets. C^4 addresses a different source of critic instability, controlling harmful TD cross covariance with clustered replay and a gradient-based correction (Qiao et al., 2026). Rather than changing replay or the critic objective, we change the reward component of the update. SNR and NSR analyses of stochastic RL gradients provide a related optimization perspective (Roberts & Tedrake, 2008; Han & Yang, 2026). Our critic-side decomposition distinguishes conditional reward noise from variation across full backup contexts. It does not assume that larger rewards yield more reliable updates.

Changing labels and network plasticity. Primacy-bias studies, ReDo, and plasticity injection address reduced adaptability through resets, dormant-neuron recycling, or network perturbations (Nikishin et al., 2022; Sokar et al., 2023; Nikishin et al., 2023). Our curriculum keeps transitions fixed but changes reward labels. We relabel sampled transitions with the current reward first. Optional rejuvenation then responds to reward drift, sensitivity, and dormancy. This separates target correction from network recovery. Appendix A discusses the broader connections to non-stationarity and return-conditioned learning.

3 PRELIMINARIES

TD learning setup. Consider a discounted Markov decision process with state space $\mathcal{S}$, action space $\mathcal{A}$, discount $\gamma \in (0, 1)$, and a fixed dataset of N transitions, $\mathcal{D} = \{(s_i, a_i, r_i, s'_i, d_i)\}_{i=1}^{N}$, where $d_i \in \{0, 1\}$ indicates termination. For a sampled transition $z = (s, a, r, s', d)$, let Q_θ be the current critic and $V_{\text{boot}}(s')$ the base learner's target value. This notation includes its target-network and action-selection rules. It does not require a separate value network. The TD loss gradient is

$$g_\theta(z) = -\big[r + \gamma(1 - d)V_{\text{boot}}(s') - Q_\theta(s, a)\big]\nabla_\theta Q_\theta(s, a).$$

The bracket is the TD residual: the observed reward and discounted continuation value, minus the current prediction. Only the current critic is differentiated, and the target is held fixed during this update. The factor $1 - d$ removes continuation at true termination. At fixed critic values and features, scaling r changes its contribution relative to the bootstrap term. Early in training, an inaccurate bootstrap value can dominate the update. Reward magnitude alone, however, says nothing about update variability.

Critic noise-to-signal ratio. Let $m_g = \mathbb{E}[g_\theta]$ and $V_g = \mathbb{E}\|g_\theta - m_g\|_2^2$ be the gradient mean and total variance under the learner's sampling distribution, with critic and target parameters fixed. For finite second moments and $m_g \neq 0$, define

$$\text{NSR} = \frac{V_g}{\|m_g\|_2^2}. \tag{1}$$

NSR compares variation between sampled updates with their common direction. It includes both conditional reward noise and variation in states, actions, bootstrap values, and critic features. Deterministic rewards remove the first source, not the second. Appendix D.4 gives the full variance decomposition. The distinction matters because reliable reward labels alone do not determine the variability of a critic update.

Local descent and reward calibration. For an L-smooth, frozen-target TD loss and an unbiased gradient estimate from a minibatch of size B, assume variance V_g/B and nonzero mean. The descent bound in Appendix D.5 has a positive descent coefficient for learning rates

$$0 < \eta < \frac{2}{L(1 + \text{NSR}/B)}. \tag{2}$$

A lower NSR permits a larger step in this bound, but reward calibration need not lower NSR. Positive scaling changes both the reward mean and its noise, leaving reward-only label-SNR unchanged under the conditioning in Appendix D.4. A fixed potential difference changes the update's mean and can also change its variation across contexts. These observations motivate controlling reward strength and content separately. They do not establish a monotone noise reduction or a convergence guarantee for the full learner.

Potential shaping and policy preservation. Potential-based shaping addresses a different question: whether the transformed reward changes the task's optimal action ordering. For a bounded state potential Φ and fixed coefficients $\alpha > 0$ and $\lambda \geq 0$, consider

$$\widetilde{r} = \alpha r + \lambda[\gamma(1 - d)\Phi(s') - \Phi(s)].$$

Under the terminal convention in Appendix D.7, this transformation preserves optimal action ordering. The identity holds for any bounded Φ and does not require an optimal value function. Choosing a potential that helps finite-time learning is the design problem in Section 4.

4 COVERAGE-AWARE REWARD CALIBRATION

We fit a fixed state potential Φ to offline trajectory returns, then use it to modify the reward supplied to the base learner. For a transition (s, a, r, s', d) at calibration epoch e,

$$\widetilde{r}_e = \underbrace{\alpha_e r}_{\text{reward scaling}} + \underbrace{\lambda_e[\gamma(1-d)\Phi(s') - \Phi(s)]}_{\text{potential signal}}, \tag{3}$$

where $\alpha_e > 0$ scales the original reward and $\lambda_e \geq 0$ controls the potential signal. The subscript e indexes coefficient changes, not positions within a trajectory. Training keeps Φ fixed and changes only these two coefficients. We first construct the potential, then analyze its information content and specify the coefficient schedule.

4.1 FROM TRAJECTORY RETURNS TO A SHARED POTENTIAL

For each stored transition, we compute the discounted return-to-go G_i within its trajectory. We center, scale, and clip these values to obtain labels $\widehat{G}_i$. The return and terminal conventions are given in Appendix D.14. The labels describe observed behavior continuations. They can differ even for the same state, whereas a state potential must assign one value.

We fit these labels jointly on a symmetric nearest-neighbor graph of standardized state embeddings (Zhou et al., 2004). Let $W_{ij} \geq 0$ be the fixed edge weight between states i and j, and let $w_i > 0$ control the influence of label i. A larger w_i makes the fit follow that label more closely. A smaller w_i allows more influence from neighboring states. For fitted values $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_N)$ and smoothing strength $\tau \geq 0$, we solve

$$\widehat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \underbrace{\frac{1}{2}\sum_i w_i(\phi_i - \widehat{G}_i)^2}_{\text{weighted return fit}} + \underbrace{\frac{\tau}{4}\sum_{i,j} W_{ij}(\phi_i - \phi_j)^2}_{\text{neighbor smoothing}}. \tag{4}$$

The minimization imposes $\phi_i = \phi_j$ whenever the stored states are exactly identical. All sums range over the N stored transitions. Positive weights give a unique solution. The optimality condition balances each label against neighboring fitted values: an isolated state retains its label, while a connected state borrows from its neighbors (Lemma 4). Appendix D.11 gives the linear system and exact duplicate-state reduction. With graph edge set $\mathcal{E}$, a sparse conjugate-gradient solve costs $O(N + |\mathcal{E}|)$ per iteration. Clipping and fixed convex interpolation give one bounded state function Φ, independent of critic predictions.

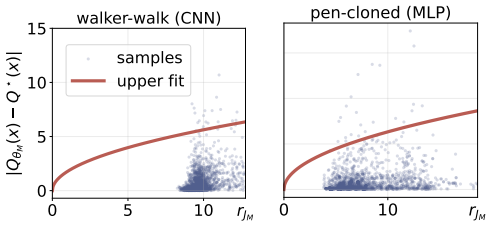

Figure 1: Local regression error versus neighborhood radius for Walker-Walk (CNN, left) and Pen-cloned (MLP, right). Points are sampled errors, and red curves show the fitted upper trends. The diagnostic illustrates the geometry term in Proposition 5. The fitted curves are empirical references, not certified bounds on the potential's error.

Geometry-based label weights. A fixed number of neighbors can be close to a state–action pair in one region and far from it in another. We use this difference to set label weights. Fix the embedding, critic family, and neighbor count J. For $x = (s, a)$, write $\bar{d}(x)$ for the mean neighbor distance and $\chi(x)$ for the local geometry factor. Our score is

$$U(x) = C_1\sqrt{\chi(x)\bar{d}(x)} + C_2\bar{d}(x) + C_0, \tag{5}$$

where $C_0 = \sigma\sqrt{2\log(2N/\delta_U)/J}$ is the common noise term, with noise scale σ and confidence parameter $\delta_U \in (0, 1)$. All constants are frozen before TD learning.

Only the geometry terms vary across queries. For MLP critics, χ uses embedding norms, and for CNN critics, it uses a frozen local kernel-size proxy. Appendix D.10 gives both forms with their full

indices. Figure 1 relates local fitting error to neighborhood radius in the two settings. It motivates including geometry in the score without treating $U(x)$ as an observed return-prediction error.

The potential must depend on the state alone, so we combine scores from nearby observed actions. Taking their maximum reduces a label's influence when a nearby behavior action has a large support score. An inverse-square map converts this score into a positive fitting weight. The local bound assumes fixed-target regularity and independent centered noise, whereas trajectory returns can be correlated. The weights are therefore geometric proxies. Section 5.1 tests them against uniform weights.

Let $\mathcal{N}_i$ index the k nearest offline states to s_i, including i. We set

$$U_i = \max_{j \in \mathcal{N}_i} U(s_j, a_j), \qquad w_i = (U_i^2 + \varepsilon)^{-1}. \tag{6}$$

The floor $\varepsilon > 0$ keeps every weight finite. The two neighborhoods play different roles: J sets the state–action neighborhood used to compute $U(x)$, while k sets the state neighborhood used to aggregate these scores. Both are fixed during potential fitting.

4.2 Potential quality and early TD information

A useful potential makes future outcomes informative at the current decision. Fix a Markov behavior policy μ, with value V^μ and advantage $A^\mu(s,a) = Q^\mu(s,a) - V^\mu(s)$. Write $c_e = \lambda_e/\alpha_e$ and measure potential error in original reward units as $\epsilon_e(s) = c_e\Phi(s) - V^\mu(s)$. The value-potential interpretation of shaping (Schulman et al., 2016) gives

$$\mathbb{E}[\widetilde{r}_e/\alpha_e \mid s,a] - A^\mu(s,a) = \gamma\mathbb{E}[(1-d)\epsilon_e(s') \mid s,a] - \epsilon_e(s). \tag{7}$$

Both expectations are over an MDP transition with the fitted potential fixed. If $c_e\Phi = V^\mu$, the expected shaped reward equals the behavior advantage. For an imperfect potential, the error in the difference between two actions is at most $2\gamma\|\epsilon_e\|_\infty$: the current-state error cancels, leaving only successor-value error. This relates potential accuracy directly to information available in a one-step TD target.

Bias and variance of the graph fit. Fix the graph, weights, smoothing strength, preprocessing, interpolation, and evaluation design. Let $\widehat{A}_e$ denote the expected normalized shaped reward produced by the linear graph fit before final potential clipping. It averages over an MDP transition, as above, but remains random through the fitted return labels. In this paragraph, $\mathbb{E}$ averages over those labels conditional on the fixed design. For a common evaluation distribution ν over state–action pairs, write $\|u\|_\nu^2 = \mathbb{E}_{(s,a)\sim\nu}[u(s,a)^2]$ and $R_e = \mathbb{E}\|\widehat{A}_e - A^\mu\|_\nu^2$. Then

$$R_e = \underbrace{\|\mathbb{E}\widehat{A}_e - A^\mu\|_\nu^2}_{\text{squared bias}} + \underbrace{\mathbb{E}\|\widehat{A}_e - \mathbb{E}\widehat{A}_e\|_\nu^2}_{\text{propagated label noise}}. \tag{8}$$

Theorem 1 expands both terms through the graph smoother and the MDP transition operator. The first includes smoothing, preprocessing, and label bias. The second retains correlations between returns. Smoothing can share reliable outcomes across states, but mixing states with different values adds bias. The signal $\widehat{A}_e$ is an analysis quantity, not a replacement for the learner's advantage estimator.

When weighting helps. Consider a connected component with labels $y_1, \dots, y_m$ of common conditional mean and uncorrelated noise with finite variances $\sigma_i^2 > 0$. Full smoothing gives the weighted mean $\widehat{u}_w = \sum_i w_i y_i / \sum_i w_i$, compared with the uniform mean $\widehat{u}_{\text{unif}} = m^{-1}\sum_i y_i$. Suppose $a/\kappa \le w_i\sigma_i^2 \le a\kappa$ for some $a > 0$ and $\kappa \ge 1$. Theorem 2 gives

$$\frac{\mathrm{Var}(\widehat{u}_w)}{\mathrm{Var}(\widehat{u}_{\text{unif}})} \le \frac{\kappa^2}{H}, \tag{9}$$

where $H = m^{-2}(\sum_i \sigma_i^2)(\sum_i \sigma_i^{-2}) \ge 1$ measures variance heterogeneity, and κ measures departure from inverse-variance weighting. All moments are conditional on the fixed design. Thus $H > \kappa^2$

guarantees a variance improvement. Across uncorrelated components satisfying the same conditions, weighted and uniform fits have identical means, and the smaller covariance also gives no larger advantage-signal risk. For equal label variances, $H = 1$ and uniform weights already achieve the minimum variance among unbiased linear averages. Exact inverse-variance weighting has $\kappa = 1$, so its benefit grows with variance heterogeneity. Smoothing is what allows those weights to matter: without graph coupling, a free parameter at each distinct state simply matches its label. Proposition 3 quantifies the finite-smoothing correction. Appendix D.2 gives a two-step MDP with an exact comparison for every $\tau > 0$: an ideally calibrated fit uses a more precise neighbor to reduce action-gap error.

Connection to early value learning. Under synchronous exact tabular policy evaluation, a zero-initialized critic for the normalized shaped reward is equivalent to an original-reward critic initialized at $c_e\Phi$. After k Bellman updates, its error in original value units is at most $\gamma^k\|\epsilon_e\|_\infty$ (Proposition 4). Potential accuracy reduces the initial error while retaining the contraction factor γ. In the two-step example, the potential can reveal the delayed action gap in the first update, before an unshaped backup reaches the outcome.

Applying the weighting comparison to geometric scores requires alignment with label precision, beyond the local geometry bound. Appendices D.1.1 and D.14 address behavior mixtures and incomplete returns. Appendices D.1.6 and D.1.7 handle postprocessing and coefficient drift. The fixed-potential, tabular result explains an initialization mechanism, not convergence of neural control.

4.3 Policy preservation and training

Proposition 1 (Fixed-epoch action-ordering invariance)**.** *Let Q^π and $\widetilde{Q}^\pi_e$ be the action values under r and $\widetilde{r}_e$. For bounded Φ, $\alpha_e > 0$, and the stated terminal convention, $\widetilde{Q}^\pi_e(s,a) = \alpha_e Q^\pi(s,a) - \lambda_e\Phi(s)$. Thus optimal action ordering is unchanged at fixed coefficients (Proposition 6).*

Policy preservation follows from the potential-difference form, independently of estimation quality. Section 4.2 distinguishes the information supplied by different potentials within this class. Appendix D.4 analyzes their contribution to conditional critic-update moments.

Annealing the calibration. The shared potential summarizes behavior-data outcomes and remains fixed as the critic learns. We retain its guidance early and gradually reduce its contribution, so later updates use rewards closer to the original task. With minibatch size B, dataset size N, and initial coefficients $\alpha_0 > 1$, $\lambda_0 > 0$, both changes follow the same schedule:

$$\underbrace{\beta_e = \exp(-Be/N)}_{\text{shared decay}}, \qquad \lambda_e = \lambda_0\beta_e, \qquad \alpha_e = 1 + (\alpha_0 - 1)\beta_e. \tag{10}$$

Each calibration epoch contains K_e TD updates with fixed coefficients. At its end, we multiply β_e by $\exp(-B/N)$. Thus e counts coefficient changes, not minibatches, and the actual number of preceding updates is $\sum_{\ell<e} K_\ell$.

The factor B/N is motivated by how quickly uniform sampling exposes the dataset, but the epoch lengths K_e control the actual decay rate and remain hyperparameters. This training clock is separate from the fixed geometric support used to weight return labels. As $e \to \infty$, $(\alpha_e, \lambda_e) \to (1, 0)$. A finite run need not reach the original reward exactly. Its remaining intervention is determined by the final β_e. Appendix D.13 gives the sampling calculation and alternative clocks.

Each epoch remains in the action-order-preserving class of Proposition 1. Across epochs, changing the reward defines a training curriculum and changes the optimization path. We implement the schedule by reevaluating sampled transitions under the current reward. Within epoch e, each sampled transition uses the calibrated TD target

$$Y_e = \widetilde{r}_e + \gamma(1 - d)V_{\text{boot}}(s'). \tag{11}$$

The reward comes from Eq. (3). The term V_{boot} retains the bootstrap rule defined in Section 3. This interface applies to TD3+BC and ReBRAC, IQL and IDQL, and CQL and SCQ. Policy extraction and regularizers retain their form, although scaling can change their effective weighting (Appendix D.12).

Optional critic recovery. The critic may lag behind changing reward labels. Reward-drift-aware relabeling and rejuvenation (RDRR) provides optional recovery: drift sets its intensity, dormancy identifies low-activity units, and reward sensitivity selects affected parameter groups. Appendix E.1 gives the algorithm. Appendix E.4 bounds tracking error assuming approximate contraction of relabelled fitting and selective rejuvenation. Fixed-coefficient action ordering holds independently of recovery.

Algorithm 1 summarizes the calibration procedure without optional RDRR.

Algorithm 1 Coverage-aware reward calibration

Require: Dataset $\mathcal{D}$, base learner, scales α_0, λ_0, epoch lengths $\{K_e\}$

```
Compute standardized return labels Ĝ_i and frozen weights w_i
Fit and bound Φ using Eq. (4)
for calibration epochs e = 0, 1, 2, ... do
    Set α_e, λ_e using Eq. (10)
    for K_e base-learner updates do
        Sample a minibatch from D
        Recompute its rewards r̃_e using Eq. (3)
        Apply the base learner's update rule with the current rewards
    end for
end for
```

5 EXPERIMENTS

We ask whether reward calibration improves the base learner, which parts of the construction contribute to the gains, and whether the same reward interface works with other learners. A separate diagnostic checks whether the fitted potential retains information about observed future returns. These tests distinguish training performance from the properties of the potential.

We use TD3+BC (Fujimoto & Gu, 2021) as the main base learner. The broad comparison covers 28 D4RL datasets: 8 AntMaze, 8 Adroit, and 12 MuJoCo locomotion datasets (Fu et al., 2020; Kim et al., 2025). These test sparse navigation, dexterous manipulation, and dense-reward control. Five V-D4RL Walker-Walk datasets provide a pixel-observation evaluation (Lu et al., 2023). The additional controls include Pen-human. Appendix B.1 describes the datasets.

Table 1 combines published scores and reruns, using the comparison suite of PARS (Kim et al., 2025). It provides broad context, not a cross-code significance test. The construction ablation and IQL/CQL controls instead use matched code and training budgets within each comparison. Scores are D4RL normalized returns, and gains below are score-point differences. Matched controls report point estimates without seed-level uncertainty. We interpret their differences descriptively rather than as evidence of statistical significance.

5.1 PERFORMANCE AND COMPONENT ANALYSIS

Table 1 places calibrated TD3+BC close to or above PARS on many tasks, while showing that it does not lead every baseline: ReBRAC scores higher on Pen-expert and Hammer-expert, and EDAC scores higher on Door-cloned. The task-level pattern is also uneven. On AntMaze-large-diverse, the reported TD3+BC and calibrated scores are 42.6 and 90.2, whereas PARS reaches 83.2. On HalfCheetah-medium-replay, the corresponding scores are 48.9, 64.5, and 57.0. By contrast, Relocate-cloned remains low at 1.6 despite exceeding the reported TD3+BC score of 0.5. Calibration does not remove the difficulty of learning from weak behavior data. These are descriptive comparisons across sources, not matched effect estimates.

The matched comparison in Table 2 more directly isolates the effect of calibration. Full calibration raises the score from 61.4 to 94.7 on AntMaze-medium-diverse, from 82.0 to 109.0 on Pen-cloned, and from 77.2 to 96.5 on Walker2d-medium-replay.

The gains have different sources across tasks. Scaling alone accounts for most of the improvement on AntMaze-medium-diverse, where the full method adds another 2.6 points. On Pen-cloned and

Table 1: D4RL normalized returns. **TD3+BC+Cal. (Ours)** adds coverage-aware reward calibration to TD3+BC and attains the highest reported average in all three domains. Baselines combine published scores and reruns. For Ours, ± is the task-level standard deviation.

Dataset	TD3+BC	IQL	CQL	SAC-N	EDAC	MCQ	MSG	SPOT	SAC-RND	ReBRAC	PARS	TD3+BC+Cal.
AntMaze												
antmaze-u	90.6	87.5	74.0	0.0	0.0	27.5	97.9	93.5	97.0	97.8	93.8	**98.4**±0.8
antmaze-u-d	82.7	62.2	84.0	0.0	0.0	0.0	79.3	40.7	66.0	88.3	89.9	**94.3**±2.5
antmaze-m-p	63.5	71.2	61.2	0.0	0.0	0.0	85.9	74.7	38.5	84.0	91.2	**91.6**±2.8
antmaze-m-d	61.4	70.0	53.7	0.0	0.0	0.0	84.6	79.1	74.7	76.3	92.0	**94.7**±2.0
antmaze-l-p	45.8	39.6	15.8	0.0	0.0	0.0	64.3	35.3	43.9	60.4	84.8	**86.0**±3.5
antmaze-l-d	42.6	47.5	14.9	0.0	0.0	0.0	71.2	36.3	45.7	54.4	83.2	**90.2**±4.0
antmaze-ultra-p	40.8	13.3	16.1	0.0	0.0	0.0	0.6	4.4	20.6	22.4	66.4	**68.0**±3.9
antmaze-ultra-d	24.6	14.2	6.5	0.0	0.0	0.0	1.0	12.0	10.5	0.8	51.4	**60.9**±7.1
Average	56.5	50.7	40.8	0.0	0.0	3.4	60.6	47.0	49.6	60.6	81.6	**85.5**
Adroit												
pen-cloned	82.0	77.2	39.2	64.1	68.2	35.3	65.6	15.2	2.5	91.8	107.5	**109.0**±12.0
pen-expert	150.2	133.6	107.0	87.1	-1.5	121.2	119.9	117.3	45.4	**154.1**	152.7	153.8±0.8
door-cloned	0.8	0.8	0.4	-0.3	**9.6**	0.2	1.1	0.0	0.2	1.1	4.3	8.8±4.5
door-expert	98.2	105.3	101.5	-0.3	106.3	73.0	83.3	0.2	73.6	104.6	106.0	**107.1**±0.3
hammer-cloned	4.8	1.1	2.1	0.2	0.3	5.2	0.5	2.5	0.1	6.7	23.3	**24.5**±18.0
hammer-expert	129.8	129.6	86.7	25.1	28.5	75.9	103.3	86.6	24.8	**133.8**	133.5	133.6±0.3
relocate-cloned	0.5	0.2	-0.1	0.0	0.0	-0.1	0.0	-0.1	0.0	0.9	1.2	**1.6**±0.4
relocate-expert	106.4	106.5	95.0	-0.3	71.9	82.5	59.3	0.0	3.4	106.6	110.5	**111.2**±1.1
Average	71.6	69.3	54.0	22.0	35.4	49.2	54.1	27.7	18.8	75.0	79.9	**81.2**
MuJoCo												
halfcheetah-r	24.8	13.1	17.5	28.0	28.4	28.5	27.2	23.8	29.0	29.5	30.4	**31.0**±0.5
hopper-r	8.0	7.9	7.9	31.3	25.3	**31.8**	31.0	31.2	31.3	8.1	25.4	31.6±0.6
walker2d-r	15.6	5.4	5.1	21.7	16.6	17.0	2.2	5.3	21.5	18.4	21.8	**22.4**±0.2
halfcheetah-m	59.8	47.4	44.0	**67.5**	65.9	64.3	60.5	58.4	66.6	65.6	64.2	67.1±0.7
hopper-m	96.0	66.3	58.5	100.3	101.6	78.4	103.5	86.0	97.8	102.0	104.1	**105.0**±0.4
walker2d-m	82.4	78.3	72.5	87.9	92.5	91.0	92.4	86.4	91.6	82.5	97.3	**98.1**±1.8
halfcheetah-m-r	48.9	44.2	45.5	63.9	61.3	56.8	52.5	52.2	54.9	51.0	57.0	**64.5**±0.4
hopper-m-r	92.7	94.7	95.0	101.8	101.0	101.6	103.7	100.2	100.5	98.1	103.1	**104.2**±0.5
walker2d-m-r	77.2	73.9	77.2	78.7	87.1	91.3	95.6	91.6	88.7	77.3	95.8	**96.5**±1.0
halfcheetah-m-e	98.2	86.7	91.6	107.1	106.3	87.5	94.2	86.9	**107.6**	101.1	103.0	107.2±1.7
hopper-m-e	104.6	91.5	105.4	110.1	110.7	111.2	111.2	99.3	109.8	107.0	113.1	**113.8**±0.2
walker2d-m-e	111.0	109.6	108.8	**116.7**	114.7	114.2	109.3	112.0	105.0	111.6	111.8	116.4±0.5
Average	68.3	59.9	60.8	76.3	76.0	72.8	73.6	69.4	75.4	71.0	77.3	**79.8**

Walker2d-medium-replay, the margins over scaling alone are 22.5 and 15.8 points. The potential therefore contributes beyond reward magnitude in these settings. Graph smoothing with uniform weights also outperforms the no-graph RTG variant on all three tasks. Geometry weighting adds 4.2, 3.1, and 1.9 points over the uniform graph. These comparisons support the combined construction, not the claim that any smooth potential helps: on AntMaze, the uniform graph scores below scaling alone.

The same reward interface also improves all twelve tested IQL and CQL learner–dataset pairs (Kostrikov et al., 2022; Kumar et al., 2020). In same-code controls, IQL gains range from 5.8 to 23.8 points and CQL gains from 3.9 to 11.7 points. For example, IQL improves from 25.7 to 49.5 on AntMaze-large-play, whereas CQL improves from 14.0 to 17.9 on the same task. The benefit thus extends beyond TD3+BC, but its size depends on the learner. Full paired scores and the separate literature references appear in Appendix B.4, Table 3. As explained in Appendix D.12, keeping the update rules does not keep all effective optimization weights unchanged.

Table 2: Construction ablation under matched TD3+BC code, budget, and seeds. Columns separate scaling-only controls from potential variants. Entries are normalized-score point estimates without seed-level uncertainty.

	Without a potential		**With a potential**		
Dataset	No calibration	Scaling only $\lambda_0 = 0$	RTG, no graph $\tau = 0$	Uniform graph weights	**Geometry weights (ours)**
AntMaze-medium-diverse	61.4	92.1	72.4	90.5	**94.7**
Pen-cloned	82.0	86.5	99.3	105.9	**109.0**
Walker2d-medium-replay	77.2	80.7	87.4	94.6	**96.5**

5.2 Schedule sensitivity and potential information

Figure 2 compares the reported schedule settings. The highlighted $\exp(-B/N)$ setting gives the highest score on four of the five tasks: AntMaze medium-play reaches 91.6, AntMaze medium-diverse 94.7, pen-human 101.7, and pen-cloned 109.0. On halfcheetah-random, the best score is 31.8 at the 0.99 setting, compared with 31.0 for $\exp(-B/N)$. The settings 0 and 1.0 score below the best intermediate setting on each task. These results characterize sensitivity to the plotted settings rather than validate the effective-exposure relation, which is analyzed separately in Appendix D.13.

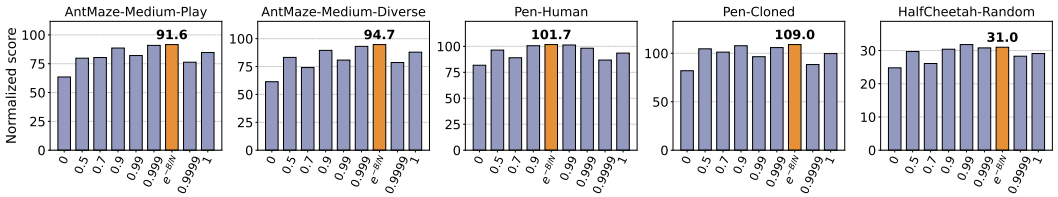

Figure 2: Sensitivity to the reported schedule setting. The highlighted $\exp(-B/N)$ setting gives the highest score on four of five tasks and remains close to the best HalfCheetah-random result.

A separate diagnostic checks what information the fixed potential retains. For each dataset, we evaluate states excluded from the reference set, sort them by $\Phi(s)$, and split them into three equal-sized groups. Mean original RTG increases from the low- to the high-potential group on all five datasets (Appendix B.4, Table 5). Adroit rank correlations range from 0.7074 to 0.8962. AntMaze correlations are only 0.0563 and 0.0684, so those results support group separation rather than accurate global ranking. Reference-set exclusion alone does not establish an independent trajectory-level test. Calibration therefore uses coarse local structure rather than requiring an exact global ordering. The split limitations are recorded with the full diagnostic.

Appendix B.4 reports the paired IQL and CQL scores and the full potential-return diagnostic, including evaluation-group sizes and high-to-low RTG ratios. Appendix B.5 adds the initial reward-scale sweep, V-D4RL Walker-Walk scores, and RDRR training curves. These results provide more detail on learner transfer, reward strength, visual observations, and recovery under changing reward labels. Together, they show how this calibration interface behaves beyond the benchmark table.

6 Conclusion

Coverage-aware reward calibration injects offline outcomes into TD targets through a fixed graph potential and annealed transformation. We connect potential quality to behavior advantages and characterize when calibrated weights reduce graph-smoothing noise. A two-step example links reliable neighbors to the first-update action gap. Ablations across learners support smoothing and geometry weighting beyond reward scaling. The potential separates return groups despite weak AntMaze ranking. Dependence on informative returns and local geometry motivates better uncertainty calibration and cheaper preprocessing. Robustness to noisy return labels also remains to be established. Outcome-based potentials can thus assist value learning while retaining the base learner's policy constraints and update rules.

REPRODUCIBILITY STATEMENT

Code is provided in the supplementary material. The main text specifies the reward transformation, potential construction, and training schedule. Appendix B describes the benchmarks, score normalization, matched controls, and return diagnostic. Appendix D.14 details potential construction. Appendices D and E provide the supporting analyses and RDRR algorithm. Appendix D.2 gives the two-step MDP and its closed-form calculations. Statistical reporting conventions accompany the corresponding results.

REFERENCES

Johannes Ackermann, Takayuki Osa, and Masashi Sugiyama. Offline reinforcement learning from datasets with structured non-stationarity. In *Reinforcement Learning Conference*, 2024.

Suzan Ece Ada, Georg Martius, Emre Ugur, and Erhan Oztop. Forecasting in offline reinforcement learning for non-stationary environments, 2025. URL https://arxiv.org/abs/2512.01987.

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

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 2015. URL https://pubsonline.informs.org/doi/10.1287/opre.2015.1408.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Nonstationary reinforcement learning: The blessing of (more) optimism. *Management Science*, 2023. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2023.4704.

Ankur Deka, Changliu Liu, and Katia P. Sycara. ARC - actor residual critic for adversarial imitation learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1446–1456. PMLR, 2023. URL https://proceedings.mlr.press/v205/deka23a.html.

Sam Michael Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012. URL https://aamas.csc.liv.ac.uk/Proceedings/aamas2012/papers/2C_3.pdf.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. RvS: What is essential for offline RL via supervised learning? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=S874XAIpkR-.

Songtao Feng, Ming Yin, Ruiquan Huang, Yu-Xiang Wang, Jing Yang, and Yingbin Liang. Non-stationary reinforcement learning under general function approximation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/feng23e.html.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning, 2020. URL https://arxiv.org/abs/2004.07219.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a8166da05c5a094f7dc03724b41886e5-Abstract.html.

Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline RL through ensembles, and why their independence matters. In *Advances in Neural Information Processing Systems*, volume 35, pp. 18267–18281, 2022.

Chengyang Gu, Yuxin Pan, Hui Xiong, and Yize Chen. STO-RL: Offline RL under sparse rewards via LLM-guided subgoal temporal order, 2026. URL https://arxiv.org/abs/2601.08107.

Haoyu Han and Heng Yang. Non-uniform noise-to-signal ratio in the reinforce policy-gradient estimator, 2026. URL https://arxiv.org/abs/2602.01460.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies, 2023. URL https://arxiv.org/abs/2304.10573.

Adam Jelley, Trevor McInroe, Sam Devlin, and Amos Storkey. Efficient offline reinforcement learning: First imitate, then improve, 2024. URL https://arxiv.org/abs/2406.13376.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f7efa4f864ae9b88d43527f4b14f750f-Abstract.html.

Jeonghye Kim, Yongjae Shin, Whiyoung Jung, Sunghoon Hong, Deunsol Yoon, Youngchul Sung, Kanghoon Lee, and Woohyung Lim. Penalizing infeasible actions and reward scaling in reinforcement learning with offline data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30769–30790. PMLR, 2025. URL https://proceedings.mlr.press/v267/kim25ai.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL https://arxiv.org/abs/2005.01643.

Jianxiong Li, Xianyuan Zhan, Haoran Xu, Xiangyu Zhu, Jingjing Liu, and Ya-Qin Zhang. When data geometry meets deep function: Generalizing offline reinforcement learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lMO7TC7cuuh.

Mingxuan Li, Junzhe Zhang, and Elias Bareinboim. Automatic reward shaping from confounded offline data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36765–36793. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25dr.html.

Cong Lu, Philip J. Ball, Tim G. J. Rudner, Jack Parker-Holder, Michael A. Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations, 2023. URL https://arxiv.org/abs/2206.04779.

Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1711–1724, 2022.

Yixiu Mao, Hongchang Zhang, Chen Chen, Yi Xu, and Xiangyang Ji. Supported value regularization for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets, 2020. URL https://arxiv.org/abs/2006.09359.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/c44a04289beaf0a7d968a94066a1d696-Abstract-Conference.html.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, 2022. URL https://proceedings.mlr.press/v162/nikishin22a.html.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/75101364dc3aa7772d27528ea504472b-Abstract-Conference.html.

Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration by random network distillation. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 26228–26244. PMLR, 2023.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

Nan Qiao, Sheng Yue, Shuning Wang, Yongheng Deng, and Ju Ren. Less is more: Clustered cross-covariance control for offline RL. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=drOy5wi6Qq.

Wenjie Qiu, Guofeng Cui, Shicheng Liu, Yuanlin Duan, and He Zhu. Preference-based policy optimization from sparse-reward offline dataset. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zyLI9LEmry.

John W. Roberts and Russ Tedrake. Signal-to-noise ratio analysis of policy gradient algorithms. In *Advances in Neural Information Processing Systems*, 2008. URL https://papers.neurips.cc/paper/3511-signal-to-noise-ratio-analysis-of-policy-gradient-algorithms.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

Yutaka Shimizu, Joey Hong, Sergey Levine, and Masayoshi Tomizuka. Strategically conservative Q-learning, 2024. URL https://arxiv.org/abs/2406.04534.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/sokar23a.html.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/26cce1e512793f2072fd27c391e04652-Abstract-Conference.html.

Tongxi Wang, Zhuoyang Xia, Xinran Chen, and Shan Liu. Tracking drift: Variation-aware entropy scheduling for non-stationary reinforcement learning, 2026. URL https://arxiv.org/abs/2601.19624.

Eric Wiewiora. Potential-based shaping and Q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 2003. URL https://www.jair.org/index.php/jair/article/view/10338.

Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. Supported policy optimization for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31278–31291, 2022.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.

Yang Yue, Rui Lu, Bingyi Kang, Shiji Song, and Gao Huang. Understanding, predicting and better resolving Q-value divergence in offline-RL. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/bd6bb13e78da078d8adcabbe6d9ca737-Abstract-Conference.html.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HN0CYZbAPw.

APPENDIX CONTENTS

A Extended Related Work **15**

B Additional Experimental Details **16**

B.1 Benchmarks . . . 16

B.2 Baseline details . . . 17

B.3 Evaluation protocol . . . 17

B.4 Matched controls and potential diagnostic . . . 17

B.5 Extended Experimental Results . . . 19

C Notation Summary **21**

D Deferred Proofs and Auxiliary Analyses **22**

D.1 Potential quality and advantage information . . . 22

D.2 An analytical example: early outcome information . . . 25

D.3 Reward magnitude and conditional noise . . . 27

D.4 Moment compression and directional reliability . . . 27

D.5 NSR descent window . . . 29

D.6 Reward-calibration bridge derivations . . . 29

D.7 Safe transformation proof and boundary . . . 31

D.8 Sparse Bernoulli reward example . . . 31

D.9 Coverage-adjusted geometry and reward-calibration consequences . . . 31

D.10 State-action reliability to a fixed state potential . . . 33

D.11 Precomputed geometry-calibrated potential . . . 35

D.12 Compatibility across offline TD learners . . . 36

D.13 Schedule design from effective exposure . . . 37

D.14 Additional implementation details for reward calibration . . . 38

E Reward-Drift Component: Algorithm and Proofs **39**

E.1 Reward-Drift-Aware Relabeling and Rejuvenation . . . 39

E.2 Bellman Drift Induced by Reward Drift . . . 41

E.3 Perturbation Bound for ReDo . . . 41

E.4 Conditional Tracking Statement for RDRR . . . 42

E.5 Idealized dynamic-regret corollary . . . 43

E.6 Additional Implementation Details . . . 43

A Extended Related Work

Section 2 compares the closest reward-scaling, shaping, initialization, and recovery methods. This appendix develops additional connections to stochastic-gradient analysis, return-conditioned learning, and environmental non-stationarity.

Sparse-reward supervision. PREFORL contrasts successful trajectories with observed failures and synthetic out-of-support failures to address overestimation under sparse rewards (Qiu et al., 2026). This uses richer outcome comparisons than an immediate scalar reward, but changes the supervision objective. Our method instead retains scalar TD targets and encodes the observed returns in a state potential. STO-RL also uses potential shaping, with LLM-generated temporal subgoals supplying progress information (Gu et al., 2026). The difference is therefore how the potential is obtained, not whether one method alone preserves action ordering. Our claim remains the fixed-coefficient PBRS identity under the stated terminal convention.

Additional shaping mechanisms and reward-structure uses. Potential-based reward shaping has also been connected to Q-value initialization and dynamic shaping potentials (Wiewiora, 2003; Devlin & Kudenko, 2012). These works show that shaping can be understood not only as reward densification, but also as a way to change the effective value landscape seen by the learner while preserving optimal policies under the appropriate assumptions. ARC studies a different use of reward structure in adversarial imitation learning: because the shaped reward is differentiable, it separates the immediate reward contribution from the discounted future return and obtains an exact gradient through the immediate reward term (Deka et al., 2023). Our shaping module targets offline critic training. Standardized return-to-go labels fit a bounded graph-smoothed potential before training. The potential remains independent of subsequent critic predictions, values, and gradients. Only its scalar coefficient is annealed by the calibration-epoch schedule.

SNR and NSR as optimization measures. At a broader optimization level, signal-to-noise style ratios have been used to study the quality of stochastic RL updates. Roberts and Tedrake formulate an SNR for policy-gradient algorithms and show that it is predictive of long-term learning performance (Roberts & Tedrake, 2008), while Han and Yang show that the REINFORCE estimator's NSR can be highly non-uniform across parameter space and often increases near optima (Han & Yang, 2026). Our critic-side diagnostics distinguish reward-label effects, bootstrap terms, and variability in the full stochastic critic update.

Reward relabeling, return-conditioned learning, and preference objectives. Reward or return relabeling has been widely used to strengthen supervision when the original reward is sparse or incomplete. Hindsight Experience Replay relabels failed goal-conditioned transitions with alternative goals (Andrychowicz et al., 2017). Decision Transformer and RvS show that offline trajectories can be exploited by conditioning policies on return or goal information and solving a supervised sequence-modeling or regression problem (Chen et al., 2021; Emmons et al., 2022). Advantage-weighted methods such as AWR and AWAC use value or advantage estimates to weight behavioral actions (Peng et al., 2019; Nair et al., 2020). Recent preference-based offline RL extends this idea by replacing or supplementing scalar reward signals with preference structure, as in PREFORL's contrastive successful-versus-failure optimization for sparse-reward offline datasets (Qiu et al., 2026). We use return-to-go labels to estimate a bounded potential before critic training, and current-reward relabeling to keep Bellman targets aligned with the present reward function.

Non-stationarity, plasticity, reset, and collapse in deep RL. Non-stationary RL studies time-varying rewards and transitions through notions such as variation budgets and dynamic regret (Besbes et al., 2015; Cheung et al., 2023; Feng et al., 2023), while recent offline variants consider structured non-stationarity in fixed datasets and forecasting future environmental change (Ackermann et al., 2024; Ada et al., 2025). Recent variation-aware entropy scheduling further shows that exploration strength can be coupled to online drift proxies, such as critic drift or high-quantile TD errors, so that maximum-entropy agents explore more after environmental changes and avoid unnecessary randomness in stable phases (Wang et al., 2026). In parallel, deep RL networks can lose plasticity under non-stationary training, overfit early data, or suffer performance collapse when objectives or data distributions change. The primacy-bias literature shows that early experience can dominate

later learning and that periodic resets of parts of the network can improve adaptation (Nikishin et al., 2022). ReDo identifies dormant neurons in deep RL networks and recycles low-activity units while minimizing immediate functional disruption (Sokar et al., 2023). Plasticity-injection methods further study how network perturbations or partial resets can restore adaptability in deep RL (Nikishin et al., 2023). Our reward-drift-aware rejuvenation considers fixed dynamics: transitions stay fixed, but reward labels may change because the calibration curriculum changes blockwise or because the practical reward function is revised across phases. This makes current-reward relabeling the default response and selective rejuvenation an auxiliary response, with reset intensity determined by reward sensitivity and dormancy rather than by generic non-stationarity alone.

B Additional Experimental Details

B.1 Benchmarks

The main comparison in Section 5 follows the D4RL benchmark convention and focuses on AntMaze, Adroit, and MuJoCo, using the same benchmark domains as the PARS comparison table (Fu et al., 2020; Kim et al., 2025). The benchmark suite covers dense locomotion, sparse goal reaching, dexterous manipulation, and optional long-horizon manipulation extensions.

MuJoCo locomotion. MuJoCo locomotion tasks evaluate continuous-control policies under dense rewards. We use the standard D4RL locomotion families `HalfCheetah`, `Hopper`, and `Walker2d`, together with dataset qualities such as `random`, `medium`, `medium-replay`, and `medium-expert`. These datasets test whether a method can improve beyond behavior data while avoiding out-of-distribution action errors.

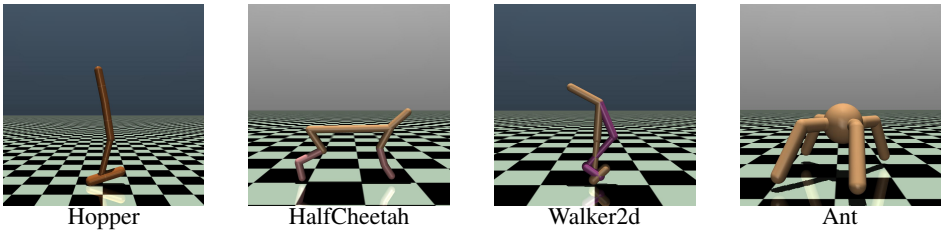

Figure 3: MuJoCo locomotion environments used for offline continuous-control evaluation.

AntMaze. AntMaze evaluates sparse-reward goal reaching with a MuJoCo ant. Because rewards are concentrated near the goal, performance depends on long-horizon credit assignment and on using the offline support effectively. We follow the standard D4RL variants `umaze`, `medium`, `large`, and `ultra`, with play and diverse data variants where applicable.

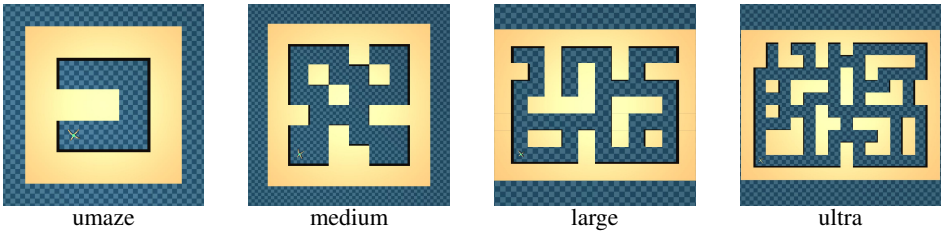

Figure 4: AntMaze layouts used to test sparse long-horizon offline RL.

Adroit. Adroit tests dexterous manipulation with a high-dimensional hand. The standard tasks are `pen`, `hammer`, `door`, and `relocate`. The main table uses `cloned` and `expert` datasets following the PARS protocol. These tasks are useful for stress-testing reward calibration because small control errors can produce large changes in contact-rich returns.

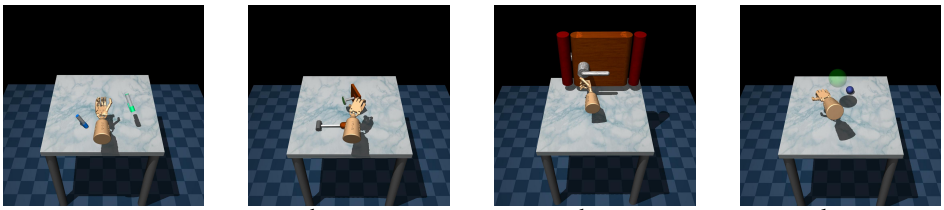

Figure 5: Adroit dexterous manipulation environments.

V-D4RL. V-D4RL extends offline RL evaluation from low-dimensional state observations to pixel observations, making representation learning part of the offline-control problem (Lu et al., 2023). In these tasks, the learner must infer control-relevant state from images using a CNN encoder while still handling the usual offline RL constraints of fixed data, distribution shift, and no additional environment interaction. We use V-D4RL Walker-Walk as an image-based complement to the state-based D4RL results.

B.2 Baseline details

The main comparison uses the same baseline suite as the PARS evaluation table (Kim et al., 2025). The numerical entries are a mix of literature-reported numbers and our own reruns, depending on the method. TD3+BC is a minimalist actor-critic method with behavior-cloning regularization (Fujimoto & Gu, 2021). IQL avoids querying out-of-distribution actions by learning an implicit value function with expectile regression (Kostrikov et al., 2022). CQL penalizes overestimated values on out-of-distribution actions (Kumar et al., 2020). SAC-N and EDAC use diversified Q-ensembles to estimate uncertainty and reduce overestimation (An et al., 2021). MCQ is a conservative Q-learning method designed to be less pessimistic while remaining robust offline (Lyu et al., 2022). MSG estimates uncertainty through ensembles and studies how ensemble independence affects pessimism (Ghasemipour et al., 2022). SPOT and SVR are support-aware policy optimization or value-regularization methods (Wu et al., 2022; Mao et al., 2024). SAC-RND uses random-network-distillation signals for anti-exploration in offline RL (Nikulin et al., 2023). ReBRAC revisits minimalist offline RL design choices and provides a strong tuned baseline (Tarasov et al., 2023). PARS combines reward scaling with infeasible-action penalization and is the closest reward-scale baseline to our method (Kim et al., 2025).

B.3 Evaluation protocol

All scores in the main table use the D4RL normalized score:

$$\text{Score} = 100 \times \frac{R - R_{\text{random}}}{R_{\text{expert}} - R_{\text{random}}}, \tag{12}$$

where R is the undiscounted return of the evaluated policy, and R_{random} and R_{expert} are the task-specific D4RL reference returns. Main-table baselines combine published values and reruns. They need not share code, seeds, or checkpoint selection. The matched controls are identified separately below. Domain averages are unweighted arithmetic means of the displayed task scores, rounded to one decimal place. Task-level standard deviations describe variation between runs. They are not averaged to estimate uncertainty in a domain mean.

B.4 Matched controls and potential diagnostic

The construction ablation in Table 2 compares TD3+BC variants under matched code, training budget, and seeds. The IQL and CQL comparisons in Table 3 use the same code and budget for each learner with and without calibration. Their literature scores are given separately in Table 4. They are not the baselines used to calculate gains. The matched controls report point estimates without seed-level standard deviations or paired confidence intervals. Their score differences are descriptive, not statistical significance tests.

Table 3: Same-code controls with IQL and CQL. Cal. denotes reward calibration, and Δ is the score-point gain over the corresponding rerun. Entries are point estimates.

	IQL			CQL		
Dataset	Rerun	+ Cal.	Δ	Rerun	+ Cal.	Δ
AntMaze-umaze	85.8	91.6	+5.8	75.3	81.1	+5.8
AntMaze-medium-play	67.9	75.4	+7.5	55.9	65.8	+9.9
AntMaze-large-play	25.7	49.5	+23.8	14.0	17.9	+3.9
Pen-human	79.5	87.0	+7.5	36.2	43.4	+7.2
Pen-cloned	76.3	94.9	+18.6	39.8	45.0	+5.2
Pen-expert	134.0	141.6	+7.6	105.9	117.6	+11.7

Table 4: Literature scores retained for context. The matched comparisons use the rerun columns in Table 3, not these values.

Dataset	**IQL (reported)**	**CQL (reported)**
AntMaze-umaze	87.5	74.0
AntMaze-medium-play	71.2	61.2
AntMaze-large-play	39.6	15.8
Pen-human	81.5	37.5
Pen-cloned	77.2	39.2
Pen-expert	133.6	107.0

Potential diagnostic. For each dataset, one fixed potential is evaluated on states excluded from its reference set. Each evaluation state is assigned the discounted return-to-go from the original, unshaped rewards. States are sorted by $\Phi(s)$ and divided into three equal-sized groups. We compare the mean return of the low-, middle-, and high-potential groups and report Spearman correlation over all evaluation states. No policy rollouts or learned return evaluator are used. Table 5 gives the group means and correlations. Table 6 gives the sample counts and high-to-low ratios. The ratios summarize separation between groups, not policy improvement, and can be large when the low-group mean is close to zero.

Table 5: Return alignment of the fixed potential. The grouped columns report mean original RTG, not normalized policy scores. All five datasets have increasing group means.

	Mean RTG by potential group			
Dataset	Low Φ	Middle Φ	High Φ	Spearman ρ
AntMaze-medium-diverse-v2	0.0093	0.0344	0.0570	0.0563
AntMaze-large-diverse-v2	0.0074	0.0397	0.1021	0.0684
Pen-human-v1	124.67	2,085.99	3,814.79	0.8962
Pen-cloned-v1	144.68	1,156.45	2,935.98	0.7879
Pen-expert-v1	736.43	1,430.39	2,380.65	0.7074

The diagnostic measures association within the offline data, not generalization to independently held-out trajectories or labels. Excluding states from the reference set does not by itself imply that their trajectories and return labels were excluded from all fitting and parameter selection. We therefore interpret the results as return alignment within the dataset.

Table 6: Evaluation sizes and high-to-low mean RTG ratios for the potential diagnostic. Evaluation states are not independent trajectories.

Dataset	Evaluation states	High/low mean RTG
AntMaze-medium-diverse-v2	994,904	6.1×
AntMaze-large-diverse-v2	994,904	13.9×
Pen-human-v1	2,927	30.6×
Pen-cloned-v1	494,216	20.3×
Pen-expert-v1	492,200	3.2×

B.5 Extended Experimental Results

The following studies examine reward scale, visual observations, and optional critic recovery. The construction ablation and schedule sweep appear in the main text (Table 2 and Figure 2). The cross-learner controls and return diagnostic appear in Appendix B.4.

Sensitivity to the initial reward scale. Figure 6 varies the initial reward scale while keeping the rest of calibration fixed. Moderate amplification performs best in this sweep: $\alpha_0 = 10$ on the two AntMaze tasks, 5 on the two Pen tasks, and 3 on HalfCheetah-random. Much larger values often reduce performance sharply. Reward magnitude therefore matters even with the potential held fixed. This is a performance comparison. The separate effects of reward scale on the mean update and its noise are analyzed in Section 3.

V-D4RL visual-control results. Table 7 extends the evaluation to V-D4RL Walker-Walk, a pixel-observation setting that uses CNN encoders. This test is complementary to the vector-state D4RL table because the reward-calibration signal must be learned through visual representations. The comparison protocol, baseline hyperparameters, and reported baseline scores are taken from the ReBRAC visual-control evaluation (Tarasov et al., 2023). The Ours row is evaluated under the same protocol. Our method obtains the best score on the medium-expert and expert datasets, with 94.9 and 93.4, respectively. On medium, it trails ReBRAC by 3.3 points. On medium-replay, it scores above ReBRAC and below Offline DV2. Offline DV2 also scores substantially higher on random.

Table 7: V-D4RL Walker-Walk results with CNN encoders, reported as mean ± standard deviation. Baselines follow the ReBRAC visual-control setup (Tarasov et al., 2023). Ours uses the same protocol. Bold means mark the highest score in each dataset.

	Walker-Walk dataset				
Method	Random	Medium-replay	Medium	Medium-expert	Expert
Offline DV2	**28.7** ± 13.0	**56.5** ± 18.1	34.1 ± 19.7	43.9 ± 34.4	4.8 ± 0.6
DrQ+BC	5.5 ± 0.9	28.7 ± 6.9	46.8 ± 2.3	86.4 ± 5.6	68.4 ± 7.5
CQL	14.4 ± 12.4	11.4 ± 12.4	14.8 ± 16.1	56.4 ± 38.4	89.6 ± 6.0
BC	2.0 ± 0.2	16.5 ± 4.3	40.9 ± 3.1	47.7 ± 3.9	91.5 ± 3.9
LOMPO	21.9 ± 8.1	34.7 ± 19.7	43.4 ± 11.1	39.2 ± 19.5	5.3 ± 7.7
ReBRAC	15.9 ± 2.3	41.6 ± 8.0	**52.5** ± 3.2	92.7 ± 1.3	81.4 ± 10.0
Ours	13.1 ± 5.0	42.3 ± 5.7	49.2 ± 2.3	**94.9** ± 3.4	**93.4** ± 6.5

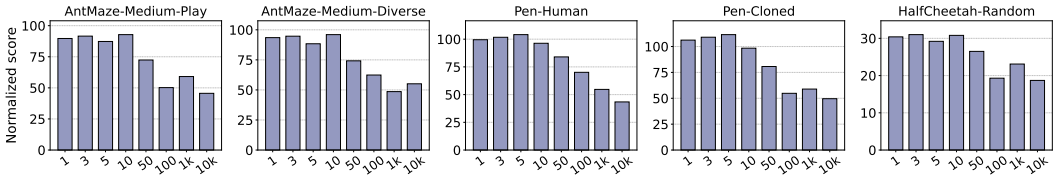

Figure 6: Sensitivity to the initial reward scale α_0. Bars report normalized scores on five D4RL tasks. Moderate amplification outperforms very large scaling in this sweep.

Ablation on RDRR. Figure 7 compares reward calibration with and without the reward-drift-aware relabeling and rejuvenation (RDRR) module from Section 4.3 and Appendix E.1. The curves report normalized score during training on four representative tasks. Across HalfCheetah-random, pen-human, pen-cloned, and AntMaze-ultra-play, adding RDRR gives higher late-stage mean performance and reduces several of the large drops seen without rejuvenation. The effect is especially visible on pen-cloned and AntMaze-ultra-play, where the non-RDRR curve is less stable. These curves support using RDRR alongside reward calibration to track changing reward labels.

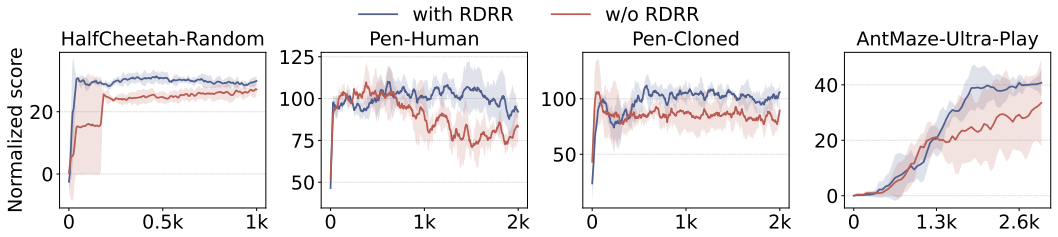

Figure 7: RDRR ablation across four tasks. Curves show three-seed mean normalized scores, with shaded regions denoting one standard deviation.

C NOTATION SUMMARY

Table 8: Notation for the reward-calibration and reward-drift components, grouped by role.

Symbol	Meaning
Data and training clock	
N, B	Number of offline transitions and minibatch size.
e, K_e, M_e	Calibration epoch, its TD-update count, and cumulative updates $M_e = \sum_{\ell<e} K_\ell$.
K_{eff}, E_e	Updates per effective exposure and $E_e = M_e/K_{\text{eff}}$.
α_e, λ_e	Reward scale and potential-signal coefficient.
β_e, q_{ep}	Shared decay $\beta_e = \exp(-Be/N)$ and per-epoch multiplier $q_{\text{ep}} = \exp(-B/N)$.
Returns, rewards, and TD targets	
$G_i, \widehat{G}_i$	Trajectory return-to-go and its centered, scaled, and clipped fitting label.
$\Phi, \widetilde{r}_e$	Fixed bounded state potential and calibrated reward at epoch e.
Y_e, V_{boot}	Calibrated TD target and the base learner's frozen bootstrap value.
$\widetilde{Q}^\pi_e, Q^\pi$	Action values under $\widetilde{r}_e$ and the original reward, respectively.
NSR	Critic gradient variance divided by squared mean, denoted $\text{NSR}_{\text{crit}}(\theta)$ in the detailed analysis.
$a_G, b_G, \epsilon^\Phi_t$	Affine return-preprocessing coefficients and trajectory discrepancy $\epsilon^\Phi_t = \Phi(s_t) - (a_G G_t + b_G)$.
Geometry: main-text notation and full appendix form	
$J, \mathfrak{m}$	Fixed neighbor count $J = J_\Phi$ and critic family (MLP or CNN).
$U(x), \chi(x), \bar{d}(x)$	Score, geometry factor, and mean distance. Full forms are $\mathcal{U}^{X,\mathfrak{m}}_\Phi$, $\widehat{\mu}^{X,\mathfrak{m}}_{J_\Phi}$, and $\bar{d}^X_{J_\Phi}$.
C_1, C_2, C_0	Geometry coefficients and common noise term $C_0 = \sigma\sqrt{2\log(2N/\delta_U)/J}$.
$\mathcal{N}_i, U_i, w_i$	State-neighbor indices $\mathcal{I}_k(s_i)$, aggregated score $\mathcal{U}^{S,\mathfrak{m}}_\Phi(s_i)$, and fitting weight.
$p(M), m_{\text{loc}}, J_M$	Distinct coverage, target local sample size, and exposure-adjusted neighbor count in the auxiliary analysis.
$\mathcal{R}_M$	Local residual estimate, not used to construct the default potential.
Potential-quality analysis	
c_e, ϵ_e	Ratio λ_e/α_e and potential error $\epsilon_e = c_e\Phi - V^\mu$.
$f, F^\sharp_e$	Appendix abbreviations for V^μ and $c_e\Phi$, giving $\epsilon_e = F^\sharp_e - f$.
$\widehat{A}_e, R_e, \nu$	Expected normalized shaped-reward signal, its mean-squared advantage error, and common evaluation distribution.
n_Φ, D_*, L_*	Distinct graph states and exactly reduced weight and Laplacian matrices.
$\mathscr{H}, \bar{y}, \Sigma$	Fixed design, conditional label mean, and conditional label covariance.
$S_\tau, T, \mathcal{B}_\gamma$	Linear smoother, fixed interpolation, and transition-difference operator $\gamma P_a - I$, with $R_e = R_{e,\tau}$ for fixed τ.
H, κ	Label-variance heterogeneity (H_{het} in the appendix) and multiplicative weight-calibration error.

D Deferred Proofs and Auxiliary Analyses

This appendix develops the potential-quality analysis and an analytical two-step example, then collects the TD-noise, geometric, and implementation details used by the method.

D.1 Potential quality and advantage information

A useful potential brings outcome information into the one-step reward. Starting from the value-potential interpretation of shaping (Schulman et al., 2016), we characterize this information through behavior-advantage error. The graph fit then admits a bias–variance analysis that identifies when calibrated weighting improves on uniform smoothing. Appendix D.2 gives a two-step MDP in which the comparison and its consequence for the first Bellman update can be calculated exactly.

D.1.1 Target and design conditioning

Fix a Markov behavior policy μ in a discounted MDP. For episodic tasks, time is included in the state when needed. Let $f = V^\mu$, $P_a u(s) = \mathbb{E}[(1-d)u(s') \mid s, a]$, and

$$(\mathcal{B}_\gamma u)(s,a) = \gamma P_a u(s) - u(s), \qquad A^\mu(s,a) = \mathbb{E}[r \mid s,a] + (\mathcal{B}_\gamma f)(s,a).$$

Terminal successors are masked. They are not zero-pinned vertices connected to the smoothing graph. A full behavior continuation started from s has expected return $f(s)$. Conditioning also on its initial action instead gives $Q^\mu(s,a)$, and conditioning on future observations can change the expectation further. A mixture of behavior policies need not admit a single Markov value target without augmenting the state.

For the matrix analysis, condition on a design $\mathscr{H}$ containing the graph, positive weights, label preprocessing constants, interpolation rule, and the evaluation operator and distribution. Let n_Φ denote the number of distinct graph states after the exact reduction in Appendix D.11. With $y \in \mathbb{R}^{n_\Phi}$ the actual node labels, including any label clipping, define

$$\bar{y} = \mathbb{E}[y \mid \mathscr{H}], \qquad \xi = y - \bar{y}, \qquad \Sigma = \mathrm{Cov}(y \mid \mathscr{H}).$$

Only finite second moments are required here. The matrix Σ may include overlapping-return and within-trajectory correlations. Writing $\bar{y} = a_G f_\mathcal{G} + b_G \mathbf{1} + b_{\mathrm{lab}}$ defines the label bias b_{lab}, with $f_\mathcal{G}$ the values at graph states. This bias is not assumed zero. Freezing a graph constructed from the same trajectories does not establish conditional unbiasedness. Independent design data and independent full continuation rollouts are one sufficient construction for an unbiased untransformed label model.

D.1.2 From potential error to advantage error

For a fixed epoch write $c_e = \lambda_e/\alpha_e$ and $F_e^\sharp = c_e\Phi$. This is the potential in original reward units. It differs from $F_e = \lambda_e\Phi$ used in the unnormalized PBRS identity. Let $r_e^\sharp = \widetilde{r}_e/\alpha_e$. The main text writes the potential error directly as $\epsilon_e = F_e^\sharp - f$ in Eq. (7). The operator notation below is its expanded matrix-analysis counterpart.

Proposition 2 (Advantage information in the shaped reward)**.** *Condition on a fitted potential and take the transition expectation under the MDP, not an empirical average over transitions reused to fit it. Then*

$$\mathbb{E}[r_e^\sharp \mid s,a] - A^\mu(s,a) = \big[\mathcal{B}_\gamma(F_e^\sharp - f)\big](s,a). \tag{13}$$

For two actions at the same state, the error in their expected shaped-reward difference is at most $2\gamma\|F_e^\sharp - f\|_\infty$.

Proof. The conditional mean of $r_e^\sharp$ is $\mathbb{E}[r \mid s,a] + \mathcal{B}_\gamma F_e^\sharp$. Subtract the defining expression for A^μ. In an action difference the current-state error cancels, leaving $\gamma(P_a - P_b)(F_e^\sharp - f)$. Each terminal-masked kernel has mass at most one, giving the stated bound. □

The identity measures information supplied to the critic: the reference is the behavior value V^μ, and the learner retains its own policy-extraction rule and value baseline.

D.1.3 BIAS AND VARIANCE OF THE GRAPH POTENTIAL

Use the reduced positive diagonal weight matrix D_* and graph Laplacian L_*, and define $S_\tau = (D_* + \tau L_*)^{-1} D_*$. For a finite evaluation state set containing all relevant successors, let T be the fixed extension from graph nodes to these states. We have $T = I$ when they coincide. Let ν be a fixed probability distribution over evaluation state–action pairs, $W_\nu = \mathrm{diag}(\nu)$, and let $\mathcal{B}_\gamma$ also denote its matrix on that set. All candidates use the same ν, MDP operator, and reward units. For fixed τ, the main-text signal is $\widehat{A}_e = \mathbb{E}[r \mid s, a] + \mathcal{B}_\gamma(c_e T S_\tau y)$, and $R_e = R_{e,\tau}$. Thus Eq. (8) separates the squared bias and variance of this signal. The next theorem gives their explicit matrix form.

Theorem 1 (Conditional advantage-signal risk)**.** *For the linear, pre-output-clipping fit $F^\sharp_{e,\tau} = c_e T S_\tau y$, define $R_{e,\tau} = \mathbb{E}[\|\mathcal{B}_\gamma(F^\sharp_{e,\tau} - f)\|_\nu^2 \mid \mathscr{H}]$. Then*

$$\begin{aligned} R_{e,\tau} = {} & \|\mathcal{B}_\gamma(c_e T S_\tau \bar{y} - f)\|_\nu^2 \\ & + c_e^2 \,\mathrm{tr}\big(W_\nu \mathcal{B}_\gamma T S_\tau \Sigma S_\tau^\top T^\top \mathcal{B}_\gamma^\top\big)\,. \end{aligned} \tag{14}$$

Proof. Substitute $y = \bar{y} + \xi$. The mixed term vanishes because $\mathbb{E}[\xi \mid \mathscr{H}] = 0$. For any fixed matrix C, $\mathbb{E}[\|C\xi\|_2^2 \mid \mathscr{H}] = \mathrm{tr}(C\Sigma C^\top)$. Apply this with $C = c_e W_\nu^{1/2} \mathcal{B}_\gamma T S_\tau$. □

The first term includes graph bias, preprocessing bias, and scale mismatch. The second describes noise propagated through both graph smoothing and MDP transitions. Setting $S_\tau = I$ gives the no-graph comparison. Replacing D_* by uniform weights gives another estimator, not another risk definition. Lower ordinary node MSE alone does not imply lower $R_{e,\tau}$.

D.1.4 WHEN NONUNIFORM WEIGHTS HELP

Consider a connected component of m graph states. Its labels have a common conditional mean u and conditionally uncorrelated noise of variances $0 < \sigma_i^2 < \infty$. These are variances of the actual input labels, not necessarily of single trajectory returns. The fully smoothed fit is $\widehat{u}_w = \sum_i q_i y_i$, with $q_i = w_i / \sum_j w_j$.

Theorem 2 (A calibrated-weight comparison)**.** *Suppose there are constants $a > 0$ and $\kappa \geq 1$ such that $a/\kappa \leq w_i \sigma_i^2 \leq a\kappa$ for every node. Define*

$$H_{\mathrm{het}} = \frac{(\sum_i \sigma_i^2)(\sum_i \sigma_i^{-2})}{m^2}. \tag{15}$$

Then

$$\mathrm{Var}(\widehat{u}_w \mid \mathscr{H}) \leq \frac{\kappa^2}{\sum_i \sigma_i^{-2}}, \qquad \frac{\mathrm{Var}(\widehat{u}_w \mid \mathscr{H})}{\mathrm{Var}(\widehat{u}_{\mathrm{unif}} \mid \mathscr{H})} \leq \frac{\kappa^2}{H_{\mathrm{het}}}. \tag{16}$$

Thus $H_{\mathrm{het}} > \kappa^2$ is a sufficient condition for a strict variance improvement. The main text writes $H = H_{\mathrm{het}}$ and suppresses conditioning on $\mathscr{H}$. With exact precision weights ($\kappa = 1$), the variance equals $(\sum_i \sigma_i^{-2})^{-1}$.

Proof. Let $Z = \sum_i w_i$. Conditional uncorrelatedness gives variance $\sum_i w_i^2 \sigma_i^2 / Z^2$. Since $w_i \sigma_i^2 \leq a\kappa$, this is at most $a\kappa / Z$. Also $Z \geq (a/\kappa) \sum_i \sigma_i^{-2}$. Combine the inequalities and divide by the uniform variance $m^{-2} \sum_i \sigma_i^2$. For exact precision weights, substitution gives the equality. Cauchy–Schwarz gives $H_{\mathrm{het}} \geq 1$. □

Corollary 1 (From component variances to signal risk)**.** *Suppose the graph has several such components, their label noises are conditionally uncorrelated across components, and each component's weighted variance is no larger than its uniform variance. The two fully smoothed estimates then have the same mean and their covariance difference is positive semidefinite in favor of weighting. For any common $c_e, T, \mathcal{B}_\gamma, \nu$, the weighted advantage-signal risk is no larger.*

Proof. On component j, the covariance is $v_j \mathbf{1}\mathbf{1}^\top$. The difference is a block diagonal matrix with blocks $(v_{j,\mathrm{unif}} - v_{j,w})\mathbf{1}\mathbf{1}^\top \succeq 0$. Both estimators preserve the component mean, so their bias terms are identical, even if that common mean is an affine transform of f. Congruence by $W_\nu^{1/2} \mathcal{B}_\gamma T$ preserves the order. Strict scalar variance improvement need not give strict signal improvement if the evaluation operator removes the affected direction. □

The comparison explains why unequal label precision can favor nonuniform smoothing. Applying it to Eq. (6) requires the geometric scores to track that precision. Equal-variance labels give $H_{\mathrm{het}} = 1$. Label correlation and value mismatch are retained in the general risk decomposition. Appendix D.2 illustrates both the variance benefit and the cost of smoothing across different values.

D.1.5 Finite smoothing and departures from local constancy

Let Π_w replace each component by its weighted average, and let $\ell_+ > 0$ be the smallest positive eigenvalue of $D_*^{-1/2} L_* D_*^{-1/2}$. Isolated nodes are unchanged. On a graph with no edges, $S_\tau = \Pi_w = I$ and we set the residual and ε_τ below to zero.

Proposition 3 (Finite-smoothing remainder)**.** *For every label vector y,*

$$\|(S_\tau - \Pi_w)y\|_{D_*} \le \frac{\|(I - \Pi_w)y\|_{D_*}}{1 + \tau\ell_+}. \tag{17}$$

Let $R_{e,\infty}$ use Π_w instead of S_τ, and define

$$\varepsilon_\tau = \frac{|c_e| \|W_\nu^{1/2} \mathcal{B}_\gamma T D_*^{-1/2}\|_2}{1 + \tau\ell_+} \sqrt{\mathbb{E}[\|(I - \Pi_w)y\|_{D_*}^2 \mid \mathscr{H}]}.$$

Then $|\sqrt{R_{e,\tau}} - \sqrt{R_{e,\infty}}| \le \varepsilon_\tau$.

Proof. Conjugation by $D_*^{1/2}$ gives $D_*^{1/2} S_\tau D_*^{-1/2} = (I + \tau D_*^{-1/2} L_* D_*^{-1/2})^{-1}$. The matrix $D_*^{1/2} \Pi_w D_*^{-1/2}$ is the Euclidean orthogonal projection onto the span of $D_*^{1/2} \mathbf{1}_C$ over components C, the kernel of the normalized Laplacian. The resolvent equals one on this kernel. On its orthogonal complement its eigenvalues are at most $(1 + \tau\ell_+)^{-1}$. This gives Eq. (17). Apply the operator norm and the reverse triangle inequality in the conditional L_2 space of errors to obtain the risk statement. □

For weighted and uniform fits, a sufficient finite-τ ordering condition is

$$\sqrt{R_{e,\infty}^{\mathrm{unif}}} - \sqrt{R_{e,\infty}^{w}} > \varepsilon_\tau^{w} + \varepsilon_\tau^{\mathrm{unif}}.$$

Each estimator has its own norm, spectral gap, and remainder. The condition is not implied by using the same numerical τ. A risk difference bound also has a mixed term: $|R_{e,\tau} - R_{e,\infty}| \le 2\sqrt{R_{e,\infty}}\varepsilon_\tau + \varepsilon_\tau^2$. For nonconstant component means, retain the first term of Eq. (14). Variance reduction alone cannot establish an ordering of the full risks.

D.1.6 Preprocessing and interpolation

At the graph nodes ($T = I$), even without smoothing or label bias, the mean potential error in original reward units is $(c_e a_G - 1)f + c_e b_G \mathbf{1}$. A constant-preserving interpolator adds $c_e a_G (T f_{\mathcal{G}} - f)$ off the graph. Under terminal masking, $\mathcal{B}_\gamma \mathbf{1}(s,a) = \gamma \Pr(d = 0 \mid s,a) - 1$. Thus centering cannot simply be dropped from the advantage-error calculation. This is distinct from PBRS policy preservation.

If numerical solution, final clipping, or another postprocessing step changes the effective potential by ΔF, then

$$|\sqrt{R(F + \Delta F)} - \sqrt{R(F)}| \le (1 + \gamma)\sqrt{\mathbb{E}[\|\Delta F\|_\infty^2 \mid \mathscr{H}]}.$$

This follows from the reverse L_2 triangle inequality and $\|\mathcal{B}_\gamma u\|_\infty \le (1 + \gamma)\|u\|_\infty$. Clipping does not generally reduce advantage risk: with $\gamma = 1/2$, $\mathcal{B}u = u_2/2 - u_1$, and $u = (1, 2)$, the error is zero before clipping to $[-1, 1]$ and $-1/2$ afterward. For a fixed convex interpolator and a target with continuity modulus ω_f in the actual embedding metric,

$$|Tz(s) - f(s)| \le \max_{j \in \mathcal{N}(s)} |z_j - f(s_j)| + \omega_f(h_s),$$

where h_s bounds neighbor distance. Without such regularity and coverage, convexity only guarantees boundedness.

D.1.7 EARLY VALUE LEARNING AND REWARD DRIFT

Proposition 4 (A fixed-policy evaluation consequence)**.** *In tabular synchronous exact policy evaluation, fix $F_e^\sharp$ and initialize the value of $r_e^\sharp$ at zero. After k Bellman updates,*

$$\widetilde{V}_k^\sharp + F_e^\sharp - f = (\gamma P_\mu)^k (F_e^\sharp - f), \qquad \|\widetilde{V}_k^\sharp + F_e^\sharp - f\|_\infty \le \gamma^k \|F_e^\sharp - f\|_\infty.$$

Proof. The shaped Bellman map satisfies $\mathcal{T}_\mu^\sharp v + F_e^\sharp = \mathcal{T}_\mu(v + F_e^\sharp)$. Subtract $f = \mathcal{T}_\mu f$ and iterate. The masked P_μ has row sums at most one. □

This gives the initialization interpretation of PBRS (Wiewiora, 2003): potential accuracy controls the initial error, while the contraction factor remains γ. The sup-norm error here and the advantage-signal risk are different criteria. For annealed coefficients, reward variation adds a tracking term. If $|r| \le R_{\max}$ and $|\Phi| \le B_\Phi$, consecutive epochs obey

$$\|\widetilde{r}_{e+1} - \widetilde{r}_e\|_\infty \le R_{\max}|\alpha_{e+1} - \alpha_e| + (1+\gamma)B_\Phi|\lambda_{e+1} - \lambda_e|.$$

This bounds the reward-variation term of Lemma 6. The optional RDRR tracking result additionally assumes approximate contraction of its fitting and recovery steps.

D.2 AN ANALYTICAL EXAMPLE: EARLY OUTCOME INFORMATION

A two-step MDP makes the role of label weighting explicit. The immediate reward contains no information about the first decision, but the offline data contain noisy observations of its outcomes. We compare raw returns, uniform graph smoothing, and ideally calibrated weighting by solving the graph fit and its action-gap error in closed form.

MDP and offline labels. Let $0 < \gamma < 1$. At s_0, actions $a \in \{+, -\}$ lead deterministically to x_a with zero reward. Each x_a has one action, which gives reward $u_a + \epsilon_{x_a}$ and terminates. A distinct auxiliary state z_a has one action, reward $u_a + \epsilon_{z_a}$, and the same terminal successor †:

$$s_0 \xrightarrow[r=0]{a} x_a \xrightarrow{r=u_a+\epsilon_{x_a}} \dagger, \qquad z_a \xrightarrow{r=u_a+\epsilon_{z_a}} \dagger, \qquad a \in \{+,-\}.$$

The behavior policy chooses the two root actions uniformly. For $\sigma > 0$ and $q \ge 1$, let the reward noises be independent, taking values $\pm\sigma$ at x_a and $\pm\sigma/\sqrt{q}$ at z_a with equal probability. Thus $V^\mu(x_a) = V^\mu(z_a) = u_a$ and the root action gap is $\Delta = Q^\mu(s_0,+) - Q^\mu(s_0,-) = \gamma(u_+ - u_-)$. Evaluation starts at s_0. The offline fitting data consist of one independently started full continuation from each of x_+, z_+, x_-, z_-. Their return labels satisfy

$$y_{x_a} = u_a + \xi_{x_a}, \qquad y_{z_a} = u_a + \xi_{z_a}, \qquad \mathrm{Var}(\xi_{x_a}) = \sigma^2, \quad \mathrm{Var}(\xi_{z_a}) = \sigma^2/q,$$

where q is the ratio of label precisions. All expectations below are exact expectations over these four independent labels, rather than averages from simulated runs.

A graph fit with two-state components. Fix an undirected unit-weight edge between x_a and z_a for each action, with no edge between branches. These edges express similarity of values, not MDP transitions. True terminal states are masked and excluded from the graph. Use identity return preprocessing and $c_e = 1$, so $r^\sharp = \widetilde{r}_e/\alpha_e = r + \gamma(1-d)\Phi(s') - \Phi(s)$. The root potential can be any fixed bounded value. Set the clipping threshold above $\max_a |u_a| + \sigma$, so clipping is inactive and no interpolation is needed.

On each component, compare uniform weights $(w_x, w_z) = (1,1)$ with ideal precision weights

$$(w_x, w_z)_{\mathrm{cal}} = \left(\frac{2}{1+q}, \frac{2q}{1+q}\right).$$

Both have mean weight one and use the same graph and τ. This is the exactly calibrated case of Theorem 2: $w_x\sigma^2 = w_z\sigma^2/q$. Solving the two-node system gives

$$\begin{pmatrix} w_x + \tau & -\tau \\ -\tau & w_z + \tau \end{pmatrix} \begin{pmatrix} \phi_{x_a} \\ \phi_{z_a} \end{pmatrix} = \begin{pmatrix} w_x y_{x_a} \\ w_z y_{z_a} \end{pmatrix},$$
$$\phi_{x_a} = (1-t)y_{x_a} + t y_{z_a}, \qquad t = \frac{\tau w_z}{w_x w_z + \tau(w_x + w_z)}. \tag{18}$$

Consequently,

$$t_{\mathrm{unif}} = \frac{\tau}{1+2\tau}, \qquad t_{\mathrm{cal}} = \frac{\tau q(1+q)}{2q+\tau(1+q)^2}. \tag{19}$$

The calibrated fit borrows more from the neighbor with the more precise return label.

Exact action-gap error. For fixed fitted labels, the difference of expected shaped rewards at s_0 is $\widehat{\Delta}_\tau = \gamma(\phi_{x_+} - \phi_{x_-})$, because the root potential cancels. This is the action-difference form of Proposition 2. Its mean-squared error is

$$\begin{aligned} R_{\mathrm{gap}}(t) := \mathbb{E}_y[(\widehat{\Delta}_\tau - \Delta)^2] &= 2\gamma^2\sigma^2\left[(1-t)^2 + \frac{t^2}{q}\right] \\ &= 2\gamma^2\sigma^2\left[\frac{1}{1+q} + \frac{1+q}{q}\left(t - \frac{q}{1+q}\right)^2\right]. \end{aligned} \tag{20}$$

This uses a common action-difference readout, rather than the state–action average defining $R_{e,\tau}$. For $q > 1$ and $0 < \tau < \infty$, Eq. (19) gives $0 < t_{\mathrm{unif}} < t_{\mathrm{cal}} < q/(1+q)$. Equation (20) is strictly decreasing on this interval, proving

$$R^{\mathrm{cal}}_{\mathrm{gap}}(\tau) < R^{\mathrm{unif}}_{\mathrm{gap}}(\tau) < R^{\mathrm{raw}}_{\mathrm{gap}}, \qquad R^{\mathrm{raw}}_{\mathrm{gap}} = 2\gamma^2\sigma^2.$$

At $\tau = 0$ both fits recover raw labels. At $q = 1$ the two weighting rules coincide. In the fully smoothed limit,

$$R^{\mathrm{unif}}_{\mathrm{gap}}(\infty) = \frac{\gamma^2\sigma^2}{2}\left(1 + \frac{1}{q}\right), \qquad R^{\mathrm{cal}}_{\mathrm{gap}}(\infty) = \frac{2\gamma^2\sigma^2}{1+q}. \tag{21}$$

Their ratio is $4q/(1+q)^2 = 1/H_{\mathrm{het}}$. For example, a precision ratio $q = 9$ gives a calibrated-to-uniform MSE ratio of $9/25$, directly from the formulas. The gain comes from where the smoothing places label influence, not from increasing the overall weight scale.

What the first Bellman update receives. Consider synchronous tabular policy evaluation with the exact MDP Bellman operator and zero initialization for both the original and normalized shaped critics. At the root, the first update is

$$Q_1(s_0, a) = 0, \qquad Q^\sharp_1(s_0, a) = \gamma\phi_{x_a} - \Phi(s_0).$$

The corresponding action-gap MSEs are Δ^2 and $R_{\mathrm{gap}}(t)$. Whenever $R_{\mathrm{gap}}(t) < \Delta^2$, the fitted potential therefore supplies a more accurate first-update action gap than the original immediate reward. After two exact updates, both recover the true gap: the shaped successor value $Q^\sharp_1(x_a) = u_a - \phi_{x_a}$ cancels the potential in the next backup. Thus the benefit is earlier access to outcome information, consistent with the initialization interpretation of PBRS (Wiewiora, 2003).

Role of local similarity and calibration. The example separates the two requirements of the construction: graph neighbors have similar values, and fitting weights reflect relative label precision. To see the first requirement quantitatively, change the auxiliary state's reward mean to $u_a + b_a$, leaving the root branches and noise variances unchanged. Keep clipping inactive by choosing its threshold above $\max_a\{|u_a| + \sigma,\ |u_a + b_a| + \sigma/\sqrt{q}\}$. The same calculation gives

$$R_{\mathrm{gap}}(t; b) = R_{\mathrm{gap}}(t) + \gamma^2 t^2 (b_+ - b_-)^2.$$

Borrowing more from a neighbor trades lower variance for greater sensitivity to value mismatch. The precision weights above idealize the second requirement. Their alignment with label uncertainty is an assumption of this example, not a consequence of geometric proximity. Theorem 2 allows imperfect calibration, while Theorem 1 retains graph bias and label correlation. Together, these results explain the statistical role intended for coverage-aware weighting without equating this analytical construction with the geometric score used in the benchmark experiments.

D.3 Reward magnitude and conditional noise

For the detailed update analysis, write $x = (s, a)$ and let y contain the next-state and any target-action arguments. We use $Q_{\bar{\theta}}(y)$ for the frozen bootstrap value denoted $V_{\mathrm{boot}}(s')$ in the main text, including the base learner's target rule. At update t, collect the critic gradients for a minibatch of size B in $\Psi_t = [\nabla_\theta Q_{\theta_t}(x_{t,1}), \ldots, \nabla_\theta Q_{\theta_t}(x_{t,B})]$. Let $\boldsymbol{q}_t, \boldsymbol{q}_t^\star \in \mathbb{R}^B$ stack the current values $Q_{\theta_t}(x_{t,b})$ and terminal-masked targets $(1 - d_{t,b})Q_{\bar{\theta}}(y_{t,b})$. Let $\mathcal{F}_t^{\mathrm{rf}}$ contain the sampled reward-free contexts, features, and frozen target values. Conditional on this information, write $\boldsymbol{r}_t = \boldsymbol{\mu}_t + \boldsymbol{\xi}_t$, with conditional mean $\boldsymbol{\mu}_t$ and centered noise of covariance R_t. The minibatch-average direction separates into a reward component, a bootstrap component, and label noise:

$$\widehat{d}_t = \underbrace{B^{-1}\Psi_t \boldsymbol{\mu}_t}_{a_t\text{: reward anchor}} + \underbrace{B^{-1}\Psi_t(\gamma \boldsymbol{q}_t^\star - \boldsymbol{q}_t)}_{c_t\text{: bootstrap component}} + \underbrace{B^{-1}\Psi_t \boldsymbol{\xi}_t}_{\text{label noise}}. \tag{22}$$

The anchor-to-bootstrap ratio is

$$\mathrm{ABR}_t = \frac{\|a_t\|_2}{\|c_t\|_2 + \varepsilon},$$

where $\varepsilon > 0$. At fixed contexts, features, and critic values, positive reward scaling changes the numerator without changing the bootstrap component. It does not improve reward-only label-SNR: the reward mean scales linearly and its covariance quadratically. ABR measures relative magnitude, not whether an update direction is useful. With deterministic rewards, conditional reward noise vanishes even though the full stochastic critic gradient can still vary. These quantities support the following analysis. The experiments do not report measured ABR, label-SNR, or critic-NSR curves.

D.4 Moment compression and directional reliability

The label-SNR and critic-NSR quantities are

$$\begin{aligned}
\mathrm{SNR}_t^{\mathrm{lab}} &:= \frac{\|a_t + c_t\|_2^2}{\mathrm{tr}(B^{-2}\Psi_t R_t \Psi_t^\top)}, \\
\mathrm{SNR}_t^{\mathrm{lab},r} &:= \frac{\|a_t\|_2^2}{\mathrm{tr}(B^{-2}\Psi_t R_t \Psi_t^\top)}, \\
\mathrm{NSR}_{\mathrm{crit}}(\theta) &:= \frac{\mathbb{E}\|g_\theta - \mathbb{E}g_\theta\|_2^2}{\|\mathbb{E}g_\theta\|_2^2}.
\end{aligned} \tag{23}$$

Label-SNR isolates conditional reward-label uncertainty, while critic-NSR measures variability of the full stochastic critic gradient. The main text abbreviates $\mathrm{NSR}_{\mathrm{crit}}(\theta)$ as NSR at fixed critic parameters. These quantities are defined for analysis, rather than used as assumptions or method inputs.

$$\mathbb{E}[\widehat{d}_t \mid \mathcal{F}_t^{\mathrm{rf}}] = a_t + c_t, \qquad \mathrm{Cov}(\widehat{d}_t \mid \mathcal{F}_t^{\mathrm{rf}}) = \frac{1}{B^2}\Psi_t R_t \Psi_t^\top. \tag{24}$$

Proof of Eq. (24). Using $\boldsymbol{r}_t = \boldsymbol{\mu}_t + \boldsymbol{\xi}_t$,

$$\widehat{d}_t = \frac{1}{B}\Psi_t(\boldsymbol{\mu}_t + \gamma \boldsymbol{q}_t^\star - \boldsymbol{q}_t) + \frac{1}{B}\Psi_t \boldsymbol{\xi}_t.$$

Taking conditional expectation gives $a_t + c_t$. Since Ψ_t is $\mathcal{F}_t^{\mathrm{rf}}$-measurable and $\mathrm{Cov}(\boldsymbol{\xi}_t \mid \mathcal{F}_t^{\mathrm{rf}}) = R_t$,

$$\mathrm{Cov}(\widehat{d}_t \mid \mathcal{F}_t^{\mathrm{rf}}) = B^{-2}\Psi_t R_t \Psi_t^\top.$$

Assume $\mathrm{tr}(\Sigma_t) > 0$. If $m_t := \mathbb{E}[\widehat{d}_t \mid \mathcal{F}_t^{\mathrm{rf}}]$ and $\Sigma_t := \mathrm{Cov}(\widehat{d}_t \mid \mathcal{F}_t^{\mathrm{rf}})$, then the reward-label directional-reliability bound used in the paper is

$$\mathbb{P}\Big(\langle \widehat{d}_t, m_t \rangle \le 0 \mid \mathcal{F}_t^{\mathrm{rf}}\Big) \le \frac{1}{1 + \mathrm{SNR}_t^{\mathrm{lab}}}. \tag{25}$$

Proof of Eq. (25). Let $Y := \langle \widehat{d}_t, m_t \rangle$. Then $\mathbb{E}[Y \mid \mathcal{F}_t^{\mathrm{rf}}] = \|m_t\|_2^2$ and

$$\mathrm{Var}(Y \mid \mathcal{F}_t^{\mathrm{rf}}) = m_t^\top \Sigma_t m_t \le \|m_t\|_2^2 \,\mathrm{tr}(\Sigma_t).$$

Cantelli's inequality gives

$$\begin{aligned}\mathbb{P}(Y \le 0 \mid \mathcal{F}_t^{\mathrm{rf}}) &\le \frac{\mathrm{Var}(Y \mid \mathcal{F}_t^{\mathrm{rf}})}{\mathrm{Var}(Y \mid \mathcal{F}_t^{\mathrm{rf}}) + \|m_t\|_2^4} \\ &\le \frac{1}{1 + \|m_t\|_2^2 / \,\mathrm{tr}(\Sigma_t)} = \frac{1}{1 + \mathrm{SNR}_t^{\mathrm{lab}}}.\end{aligned}$$

When $\mathrm{tr}(\Sigma_t) = 0$, reward-label noise vanishes and the label-SNR bound is interpreted separately. Remaining stochasticity is described by the unconditional critic NSR below.

For the population NSR decomposition, let $v = (x, y, d)$ be the reward-free backup context and $z = (v, r)$ the realized sample. Define $b_\theta(v) = \gamma(1-d)Q_{\bar\theta}(y) - Q_\theta(x)$ and write

$$r = \mu_r(v) + \xi_r, \qquad \mathbb{E}[\xi_r \mid v] = 0, \qquad \mathrm{Var}(\xi_r \mid v) = \sigma_r^2(v).$$

$$m_g(\theta) = -\mathbb{E}\big[(\mu_r(v) + b_\theta(v))\nabla_\theta Q_\theta(x)\big], \tag{26}$$

$$\begin{aligned}V_g(\theta) = \mathbb{E}\big[\sigma_r^2(v)\|\nabla_\theta Q_\theta(x)\|_2^2\big] \\ + \mathrm{tr}\,\mathrm{Cov}\big((\mu_r(v) + b_\theta(v))\nabla_\theta Q_\theta(x)\big).\end{aligned} \tag{27}$$

Derivation of Eq. (27). Since $g_\theta(v, r) = -(\mu_r(v) + \xi_r + b_\theta(v))\nabla_\theta Q_\theta(x)$ and $\mathbb{E}[\xi_r \mid v] = 0$, the mean is Eq. (26). The total variance identity, conditioned on v, yields

$$V_g(\theta) = \mathbb{E}\left[\mathrm{Var}(g_\theta(v, r) \mid v)\right] + \mathrm{tr}\,\mathrm{Cov}\left(\mathbb{E}[g_\theta(v, r) \mid v]\right),$$

which is exactly Eq. (27).

Lemma 1 (Label moments under fixed reward calibration)**.** *Fix a calibration epoch e, and condition on the sampled reward-free contexts, critic features, frozen bootstrap values, current predictions, and the precomputed potential. Let $\boldsymbol{h}_t$ collect the potential differences*

$$h_{t,i} := \gamma(1 - d_{t,i})\Phi(s'_{t,i}) - \Phi(s_{t,i}),$$

and suppose the original conditional reward law is $\boldsymbol{r}_t = \boldsymbol{\mu}_t + \boldsymbol{\xi}_t$, with $\mathbb{E}[\boldsymbol{\xi}_t \mid \mathcal{F}_t^{\mathrm{rf}}] = 0$ and $\mathrm{Cov}(\boldsymbol{\xi}_t \mid \mathcal{F}_t^{\mathrm{rf}}) = R_t$. For the fixed-epoch calibrated labels

$$\widetilde{\boldsymbol{r}}_t^{(e)} = \alpha_e \boldsymbol{r}_t + \lambda_e \boldsymbol{h}_t, \qquad \alpha_e > 0,$$

define $u_t^\Phi := B^{-1}\Psi_t \boldsymbol{h}_t$. Then the reward-anchor and conditional label-noise covariance become

$$\widetilde{a}_t^{(e)} = \alpha_e a_t + \lambda_e u_t^\Phi, \qquad \widetilde{\Sigma}_t^{(e)} = \alpha_e^2 \Sigma_t. \tag{28}$$

Consequently, when $\mathrm{tr}(\Sigma_t) > 0$,

$$\begin{aligned}\widetilde{\mathrm{SNR}}_{t,e}^{\mathrm{lab},r} &= \frac{\|\alpha_e a_t + \lambda_e u_t^\Phi\|_2^2}{\alpha_e^2 \,\mathrm{tr}(\Sigma_t)}, \\ \widetilde{\mathrm{SNR}}_{t,e}^{\mathrm{lab}} &= \frac{\|\alpha_e a_t + \lambda_e u_t^\Phi + c_t\|_2^2}{\alpha_e^2 \,\mathrm{tr}(\Sigma_t)}.\end{aligned} \tag{29}$$

In the special case $\lambda_e = 0$, positive scaling alone leaves the reward-only label SNR unchanged, although it can change the reward-anchor magnitude relative to the current bootstrap component c_t.

Proof. Because $\boldsymbol{h}_t$ is fixed under the conditioning event,

$$\mathbb{E}[\widetilde{\boldsymbol{r}}_t^{(e)} \mid \mathcal{F}_t^{\mathrm{rf}}] = \alpha_e \boldsymbol{\mu}_t + \lambda_e \boldsymbol{h}_t, \qquad \mathrm{Cov}(\widetilde{\boldsymbol{r}}_t^{(e)} \mid \mathcal{F}_t^{\mathrm{rf}}) = \alpha_e^2 R_t.$$

Multiplying the conditional mean and covariance by $B^{-1}\Psi_t$ and $B^{-2}\Psi_t(\cdot)\Psi_t^\top$, respectively, gives Eq. (28). The SNR identities follow by substituting these moments into Eq. (23). □

Full-gradient moments at fixed critic parameters. Write $\psi_\theta(x) = \nabla_\theta Q_\theta(x)$ and $h = \gamma(1-d)\Phi(s') - \Phi(s)$. Keeping the sampling distribution, critic, and potential fixed gives

$$\widetilde{m}_g^{(e)} = -\alpha_e \mathbb{E}[r\psi_\theta(x)] - \lambda_e \mathbb{E}[h\psi_\theta(x)] - \mathbb{E}[b_\theta(v)\psi_\theta(x)], \tag{30}$$

$$\widetilde{V}_g^{(e)} = \operatorname{tr}\operatorname{Cov}\big((\alpha_e r + \lambda_e h + b_\theta(v))\psi_\theta(x)\big). \tag{31}$$

The ratio $\widetilde{V}_g^{(e)}/\|\widetilde{m}_g^{(e)}\|_2^2$ is the critic NSR for nonzero mean. Calibration changes both terms, so neither the potential-quality comparison nor PBRS alone implies that this ratio decreases.

D.5 NSR DESCENT WINDOW

If a minibatch loss-gradient estimator $\widehat{g}_B$ is unbiased with conditional variance $V_g(\theta)/B$, then for an L-smooth frozen-target TD objective with decoupled target,

$$\mathbb{E}[\mathcal{L}(\theta^+) \mid \theta] \le \mathcal{L}(\theta) - \eta\left(1 - \frac{L\eta}{2}\left(1 + \frac{\mathrm{NSR}_{\mathrm{crit}}(\theta)}{B}\right)\right)\|\nabla\mathcal{L}(\theta)\|_2^2, \tag{32}$$

where $\theta^+ = \theta - \eta\widehat{g}_B(\theta)$.

Derivation of Eq. (32). By L-smoothness,

$$\mathcal{L}(\theta - \eta\widehat{g}_B) \le \mathcal{L}(\theta) - \eta\langle\nabla\mathcal{L}(\theta), \widehat{g}_B\rangle + \frac{L\eta^2}{2}\|\widehat{g}_B\|_2^2.$$

Taking conditional expectation and using unbiasedness,

$$\mathbb{E}[\mathcal{L}(\theta^+) \mid \theta] \le \mathcal{L}(\theta) - \eta\|\nabla\mathcal{L}(\theta)\|_2^2 + \frac{L\eta^2}{2}\left(\|\nabla\mathcal{L}(\theta)\|_2^2 + \frac{V_g(\theta)}{B}\right).$$

Substituting $V_g(\theta) = \mathrm{NSR}_{\mathrm{crit}}(\theta)\|\nabla\mathcal{L}(\theta)\|_2^2$ gives Eq. (32).

D.6 REWARD-CALIBRATION BRIDGE DERIVATIONS

We derive the local recursion connecting weak reward anchoring to bootstrap takeover for the default decoupled-target TD update. A separate coupled-bootstrap variant illustrates self-excitation when bootstrap values move with the online parameters.

Fix a reference set $X = \{x_i\}_{i=1}^n$, corresponding target points $Y = \{y_i\}_{i=1}^n$, and their terminal flags d_i. Within this local-recursion subsection only, evaluations and feature columns at y_i include the factor $(1-d_i)$. Thus $Q_{\bar\theta_t}(Y)$ stacks terminal-masked bootstrap values. Define the centered TD-error vector

$$\bar{u}_t := Q_{\theta_t}(X) - \boldsymbol{\mu} - \gamma Q_{\bar\theta_t}(Y), \tag{33}$$

and the local feature matrix

$$Z_t := [\nabla_\theta Q_{\theta_t}(x_1), \ldots, \nabla_\theta Q_{\theta_t}(x_n)]\,, \qquad K_t := Z_t^\top Z_t.$$

Under a local first-order expansion with the target critic held fixed during the step, $\delta_t = \xi_t - \bar{u}_t$ and the semi-gradient update can be written as

$$\theta_{t+1} = \theta_t + \eta Z_t(\xi_t - \bar{u}_t).$$

Therefore the centered TD error obeys

$$\bar{u}_{t+1} = (I - \eta K_t)\bar{u}_t + \eta K_t\xi_t + o(\eta), \tag{34}$$

where $\xi_t = \boldsymbol{r}_t - \boldsymbol{\mu}$. Freezing $K_t \equiv K$ and the target values, and retaining the first-order model, gives $A_{\mathrm{TD}} = I - \eta K$ and

$$\bar{u}_{t+1} = A_{\mathrm{TD}}\bar{u}_t + \eta K\xi_t. \tag{35}$$

For the covariance calculation, let $\mathcal{F}_t$ contain the history before drawing ξ_t, and assume $\mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0$ and $\mathrm{Cov}(\xi_t \mid \mathcal{F}_t) = \Sigma_r$. These are local noise-model assumptions, not independence supplied by repeated replay of fixed labels. With $P_t = \mathrm{Cov}(\bar{u}_t)$, the covariance follows the Lyapunov recursion

$$P_{t+1} = A_{\mathrm{TD}}P_tA_{\mathrm{TD}}^\top + \eta^2 K\Sigma_r K^\top. \tag{36}$$

Coupled-bootstrap self-excitation variant. If the bootstrap value is locally coupled to the same online parameters, rather than held fixed by a target critic during the step, the same first-order calculation introduces

$$Z_t^\star := [\nabla_\theta Q_{\theta_t}(y_1), \dots, \nabla_\theta Q_{\theta_t}(y_n)], \qquad G_t^\star := (Z_t^\star)^\top Z_t, \qquad \mathsf{S}_t := \gamma G_t^\star - K_t.$$

This coupled variant yields

$$\bar{u}_{t+1} = (I + \eta \mathsf{S}_t)\bar{u}_t - \eta \mathsf{S}_t \xi_t + o(\eta). \tag{37}$$

This recursion describes self-excitation in the coupled local model. The default update uses the decoupled-target recursion above.

To separate reward calibration from support geometry, define

$$\Pi_{\mathcal{D}}(x) := \arg\min_{\tilde{x} \in \mathcal{D}} \|x - \tilde{x}\|, \qquad d_{\mathcal{D}}(x) := \|x - \Pi_{\mathcal{D}}(x)\|. \tag{38}$$

For an envelope constant $C_{\mathrm{env}} > 0$, use the DOGE-style bound

$$\Gamma_{\mathcal{D}}(x) := C_{\mathrm{env}} \sqrt{d_{\mathcal{D}}(x)} + 2 d_{\mathcal{D}}(x), \tag{39}$$

and assume

$$|Q_\theta(x) - Q_\theta(\Pi_{\mathcal{D}}(x))| \le C_g \Gamma_{\mathcal{D}}(x). \tag{40}$$

The envelope describes support mismatch as a separate source of error. It is an analysis assumption, not a constraint enforced by reward calibration.

At the population level, use the previously defined backup context $v = (x, y, d)$ and residual $b_\theta(v)$, including the terminal mask. The decoupled TD objective is

$$\mathcal{J}(\theta) := \frac{1}{2} \mathbb{E}[(r + b_\theta(v))^2]. \tag{41}$$

With $r = \mu_r(v) + \xi$ and $\mathbb{E}[\xi \mid v] = 0$, its parameter-dependent part is

$$\mathcal{J}(\theta) = \mathrm{const} + \mathbb{E}[\mu_r(v) b_\theta(v)] + \frac{1}{2} \mathbb{E}[b_\theta(v)^2]. \tag{42}$$

Define the reward-anchor and bootstrap components

$$a(\theta) := \mathbb{E}[\mu_r(v) \nabla_\theta Q_\theta(x)], \qquad c(\theta) := \mathbb{E}[b_\theta(v) \nabla_\theta Q_\theta(x)]. \tag{43}$$

When $c(\theta) \neq 0$, the anchor-to-bootstrap ratio is

$$\mathrm{ABR}(\theta) := \frac{\|a(\theta)\|_2}{\|c(\theta)\|_2}. \tag{44}$$

If $\mathrm{ABR}(\theta) < 1$, then the update is bootstrap-dominated since

$$\langle -\nabla_\theta \mathcal{J}(\theta), c(\theta) \rangle \ge \|c(\theta)\|_2 \big(\|c(\theta)\|_2 - \|a(\theta)\|_2 \big) > 0. \tag{45}$$

Dangerous scalar mode in the coupled variant. Under the same conditional noise assumptions as Eq. (36), let A be the frozen matrix S_t of the first-order coupled model. If it has a positive real eigenvalue $\lambda_\star$ with left eigenvector $w_\star$, define

$$c_t := w_\star^\top \bar{u}_t, \qquad \zeta_t := w_\star^\top A \xi_t, \qquad \sigma_\star^2 := w_\star^\top A \Sigma_r A^\top w_\star.$$

Then

$$c_{t+1} = (1 + \eta \lambda_\star) c_t - \eta \zeta_t, \qquad \mathbb{E}[c_{t+1}^2 \mid \mathcal{F}_t] = (1 + \eta \lambda_\star)^2 c_t^2 + \eta^2 \sigma_\star^2. \tag{46}$$

Thus any nonzero initialization or injected reward noise along this mode is amplified when $1 + \eta \lambda_\star > 1$.

D.7 SAFE TRANSFORMATION PROOF AND BOUNDARY

For a bounded potential F and terminal convention $F(s_{\text{term}}) = 0$,

$$\sum_{t=0}^{\infty} \gamma^t \big(\gamma(1-d_t)F(s_{t+1}) - F(s_t)\big) = -F(s_0) \quad \text{a.s.}$$

Therefore,

$$Q^{\pi}_{kr+\gamma(1-d)F(s')-F(s)}(s,a) = kQ^{\pi}_r(s,a) - F(s).$$

Adding a constant offset c in a continuing discounted formulation gives $c/(1-\gamma)$, which is action-independent. This offset is also harmless under an absorbing-terminal convention with fixed terminal treatment. For variable-horizon episodic tasks without such a convention, a per-step constant may alter length preferences. The reward-calibration rule therefore uses the additive epoch offset $b_e^{\text{off}} = 0$, distinct from the strength ratio $c_e = \lambda_e/\alpha_e$.

An arbitrary state bias $b(s)$ is not generally safe. Although $b(s)$ is action-independent at the current state, its discounted cumulative contribution depends on the future state visitation distribution, which is policy-dependent. Thus a transformation of the form $kr + b(s)$ can change optimal action ordering unless $b(s)$ is representable as a potential-difference term plus an admissible constant.

D.8 SPARSE BERNOULLI REWARD EXAMPLE

For fixed features $\Psi_t \neq 0$ and scale $A > 0$, consider conditionally independent rewards

$$r_{t,i} = AZ_{t,i}, \qquad Z_{t,i} \sim \text{Bernoulli}(p), \qquad 0 < p \ll 1.$$

Then

$$\boldsymbol{\mu}_t = Ap\mathbf{1}, \qquad R_t = A^2 p(1-p) I_B.$$

The reward-only label SNR in Eq. (23) becomes

$$\text{SNR}_t^{\text{lab},r} = \frac{p}{1-p} \frac{\|\Psi_t \mathbf{1}\|_2^2}{\|\Psi_t\|_F^2}. \tag{47}$$

Multiplicative scaling by A changes the reward mean and covariance together, leaving this ratio unchanged. For fixed $\Psi_t \mathbf{1} \neq 0$, the sparse limit gives $\text{SNR}_t^{\text{lab},r} = \Theta(p)$. If $\Psi_t \mathbf{1} = 0$, it is zero. Positive scaling can still change the anchor-to-bootstrap magnitude, while potential shaping can add a deterministic term in critic-feature space.

D.9 COVERAGE-ADJUSTED GEOMETRY AND REWARD-CALIBRATION CONSEQUENCES

This subsection gives an auxiliary fixed-label local-regression model motivating the geometry score. It does not establish inverse-variance calibration for behavior RTGs. The quality comparison in Appendix D.1 uses separate explicit conditions. Here m_{loc} denotes the target effective local sample size, while $\mathfrak{m} \in \{\text{MLP}, \text{CNN}\}$ denotes the model family. All probabilities in this subsection are with respect to the mini-batch exposure process and the independent fixed-label noises $\{\xi_i\}_{i=1}^N$, while the offline inputs are treated as fixed.

Assumption 1 (Coverage, local geometry, and fixed-label noise)**.** The following conditions are used only for the coverage-adjusted local regression analysis.

(i) The offline dataset has N transitions. Each mini-batch draw samples a uniform mini-batch of size B, independently across draws. A fixed transition is selected in one draw with probability ρ_B, where $\rho_B = B/N$ for without-replacement mini-batches and $\rho_B = 1 - (1 - 1/N)^B$ for with-replacement draws.

(ii) For the local regression abstraction, labels satisfy $y_i = Q^\star(x_i) + \xi_i$, where the ξ_i are independent, mean-zero, σ^2-sub-Gaussian, and independent of the mini-batch sampling process.

(iii) Fix a model family $\mathfrak{m} \in \{\text{MLP}, \text{CNN}\}$. Over a radius-$r$ state-action neighborhood, the corresponding frozen NTK/CNTK feature map satisfies the local smoothness envelope

encoded by $\Lambda_{\mathfrak{m}}(x, r)$, and the target $Q^\star$ is locally $L_{\mathfrak{m}}$-regular. These properties yield the deterministic geometry contribution

$$C_{\mathfrak{m}} \sqrt{\Lambda_{\mathfrak{m}}(x, r) r} + (2C_{\mathfrak{m}} + L_{\mathfrak{m}}) r.$$

For MLP critics, $\Lambda_{\mathfrak{m}}$ is induced by the ReLU-NTK local norm factor. For CNN critics, $\Lambda_{\mathfrak{m}}$ is induced by the corresponding CNTK local size factor.

(iv) The term $\mathcal{R}_M(x, r)$ denotes the largest local fitting residual on covered samples inside the radius-r neighborhood after M mini-batch draws. It is an explicit optimization/statistical residual and is not assumed to vanish.

Let $M \geq 1$ be the cumulative number of mini-batch draws and define

$$p(M) := 1 - (1 - \rho_B)^M. \tag{48}$$

For a target effective local sample size $m_{\mathrm{loc}} \in \{1, \ldots, N\}$ and confidence parameter $\eta \in (0, 1)$, define

$$\begin{aligned} \Delta_M(\eta) &:= B \sqrt{\frac{M}{2} \log \frac{1}{\eta}}, \\ J_M^{\mathrm{raw}}(m_{\mathrm{loc}}, \eta) &:= \left\lceil \frac{m_{\mathrm{loc}} + \Delta_M(\eta) + 1}{p(M)} \right\rceil, \\ J_M(m_{\mathrm{loc}}, \eta) &:= \min\{N, J_M^{\mathrm{raw}}(m_{\mathrm{loc}}, \eta)\}. \end{aligned} \tag{49}$$

The target-effective-size condition is active when $J_M^{\mathrm{raw}}(m_{\mathrm{loc}}, \eta) \leq N$, equivalently $p(M)N \geq m_{\mathrm{loc}} + \Delta_M(\eta) + 1$. When this condition fails, the clipped index selects the full dataset for the coverage calculation. The default potential uses its own fixed neighborhood size J_Φ.

Lemma 2 (Distinct local coverage)**.** *Under Assumption 1(i), let $\mathcal{C}_M$ be the set of distinct transitions exposed after M mini-batch draws. For any fixed local neighborhood containing J offline transitions, the expected number of exposed distinct transitions in that neighborhood is $p(M)J$. Moreover, with probability at least $1 - \eta$, it is at least $p(M)J - \Delta_M(\eta)$.*

Proof. For each transition i, the probability of not being selected in one mini-batch draw is $1 - \rho_B$, hence the probability of being selected at least once after M draws is $p(M) = 1 - (1 - \rho_B)^M$. Summing the corresponding indicators over the J transitions gives the expectation. Changing one mini-batch draw can alter the distinct local count by at most B, so McDiarmid's bounded-difference inequality gives the stated lower-tail slack $\Delta_M(\eta)$. The exposed count and its expectation are nondecreasing in M. The lower bound subtracts $\Delta_M(\eta)$, which grows as $\sqrt{M}$, so the corrected bound need not be nondecreasing. The probability statement is for a fixed M and a fixed neighborhood. □

Theorem 3 (Coverage-adjusted local regression geometry bound)**.** *Fix $M \geq 1$, a query $x = (s, a)$, and $\mathfrak{m} \in \{\mathrm{MLP}, \mathrm{CNN}\}$, and suppose Assumption 1 holds. For a radius r, let $N_x(r)$ be the number of full-data samples within radius r, and define*

$$N_{\mathrm{eff}}(x, r; M, \eta) := \lfloor p(M) N_x(r) - \Delta_M(\eta) \rfloor.$$

If $N_{\mathrm{eff}}(x, r; M, \eta) \geq 1$, then with probability at least $1 - \delta - \eta$ over the exposure process and label noise,

$$\begin{aligned} |Q_{\theta_M}(x) - Q^\star(x)| \leq\ & C_{\mathfrak{m}} \sqrt{\Lambda_{\mathfrak{m}}(x, r)\, r} + (2C_{\mathfrak{m}} + L_{\mathfrak{m}}) r \\ & + \mathcal{R}_M(x, r) + \sigma \sqrt{\frac{2 \log(2N/\delta)}{N_{\mathrm{eff}}(x, r; M, \eta)}}. \end{aligned} \tag{50}$$

For this fixed query, the noise logarithm can be reduced to $\log(2/\delta)$. A simultaneous statement over multiple queries also requires allocating the coverage failure probability across their neighborhoods. The bound concerns the local regression surrogate with averaged fixed-label noise and residual $\mathcal{R}_M$.

Proof. Lemma 2 ensures that, with probability at least $1 - \eta$, the radius-r neighborhood contains at least N_{eff} distinct exposed samples. Conditioning on this covered set, the local MLP/CNN geometry condition in Assumption 1 applies to the surrogate abstraction on those N_{eff} points: the deterministic

geometry and target-regularity terms give the first two terms, the incomplete-fit term is $\mathcal{R}_M(x, r)$, and the local fixed-label surrogate noise average is bounded by the sub-Gaussian term. Combining the coverage event and the label-noise event gives the result. □

Corollary 2 (Target effective local sample size)**.** *Fix $M \geq 1$ and a query x, assume $p(M)N \geq m_{\mathrm{loc}} + \Delta_M(\eta) + 1$, and let $r_{J_M}(x)$ be the radius to the $J_M(m_{\mathrm{loc}}, \eta)$-th nearest full-data neighbor of x, with $J_M(m_{\mathrm{loc}}, \eta)$ from Eq.* (49)*. Under Assumption 1, with probability at least $1 - \delta - \eta$ over the exposure process and label noise,*

$$\begin{aligned} |Q_{\theta_M}(x) - Q^\star(x)| \leq\ & C_{\mathfrak{m}} \sqrt{\Lambda_{\mathfrak{m}}(x, r_{J_M}(x))\, r_{J_M}(x)} \\ & + (2C_{\mathfrak{m}} + L_{\mathfrak{m}}) r_{J_M}(x) + \mathcal{R}_M(x, r_{J_M}(x)) \\ & + \sigma \sqrt{\frac{2\log(2N/\delta)}{m_{\mathrm{loc}}}}. \end{aligned} \tag{51}$$

Proof. The condition $p(M)N \geq m_{\mathrm{loc}} + \Delta_M(\eta) + 1$ ensures $J_M = J_M^{\mathrm{raw}}$. By construction, $p(M)J_M(m_{\mathrm{loc}}, \eta) - \Delta_M(\eta) \geq m_{\mathrm{loc}} + 1$, so the effective distinct count inside the J_M-neighbor radius is at least m_{loc}. Substituting $r = r_{J_M}(x)$ in Theorem 3 and relaxing the noise denominator to m_{loc} proves the claim. □

Compact form of the local bound. Fix $M \geq 1$, query $x = (s, a)$, radius $\rho > 0$, and $\eta \in (0, 1)$. Write $N_{\mathrm{eff}}(x, \rho, M)$ for the corrected lower bound $N_{\mathrm{eff}}(x, \rho; M, \eta)$ from Theorem 3. Write $\widehat{Q}_M(x)$ for the local fit and $Q^\star(x)$ for its fixed regression target. $\mathcal{G}_{\mathfrak{m}}(x, \rho)$ and $\mathcal{R}_M(x, \rho)$ are the geometry and residual terms for model family $\mathfrak{m}$. The parameters σ and δ are the label-noise scale and confidence parameter.

Proposition 5 (Local error at fixed exposure)**.** *Under Assumption 1, if $N_{\mathrm{eff}}(x, \rho, M) \geq 1$, then with probability at least $1 - \delta - \eta$,*

$$\left|\widehat{Q}_M(x) - Q^\star(x)\right| \leq \underbrace{\mathcal{G}_{\mathfrak{m}}(x, \rho)}_{\textit{local geometry}} + \underbrace{\mathcal{R}_M(x, \rho)}_{\textit{fitting residual}} + \underbrace{\sigma \sqrt{\frac{2\log(2N/\delta)}{N_{\mathrm{eff}}(x, \rho, M)}}}_{\textit{label uncertainty}}. \tag{52}$$

At fixed M and η, expanding the neighborhood increases or preserves N_{eff}, at the cost of using more distant inputs in the geometry term. Corollary 2 chooses a radius that achieves a prescribed effective sample count whenever the target-size condition holds.

Remark 1 (Warm-up regime). At $M = 0$, no samples have been exposed and we take $J_0 = N$ as the initial analysis neighborhood. For $M \geq 1$ with $p(M)N < m_{\mathrm{loc}} + \Delta_M(\eta) + 1$, Eq. (49) clips J_M to N. The target-size corollary applies once its non-strict condition holds. Because the correction grows with M, the condition can fail again at large draw counts. The potential is computed with fixed J_Φ before training, and its strength follows the calibration-epoch schedule throughout. Thus the analysis index J_M does not trigger a change of potential or a restart of the curriculum.

D.10 State-action reliability to a fixed state potential

The reliability bound is stated for $x = (s, a)$, while potential-based shaping requires a single state function $\Phi(s)$. We therefore first define state-action reliability scores and then project them to states by a fixed conservative aggregation before critic training.

Let $\zeta(x)$ be a standardized state-action embedding. For a query state-action pair x, let

$$x_{(1)}(x), \ldots, x_{(J)}(x)$$

be its J nearest offline state-action neighbors in the ζ-metric. Define the average local distance and nearest-neighbor radius by

$$\bar{d}_J^X(x) := \frac{1}{J} \sum_{\ell=1}^{J} \|\zeta(x) - \zeta(x_{(\ell)}(x))\|_2, \qquad r_J^X(x) := \|\zeta(x) - \zeta(x_{(J)}(x))\|_2. \tag{53}$$

MLP instantiation. For an infinite-width two-layer ReLU MLP in the NTK regime, the induced kernel feature map satisfies the standard local Hölder-type smoothness used in the DOGE-style geometry argument. The corresponding empirical geometry factor is

$$\widehat{\mu}_J^{X,\mathrm{MLP}}(x) := \frac{1}{J}\sum_{\ell=1}^{J} \min\big\{\|\zeta(x)\|_2,\ \|\zeta(x_{(\ell)}(x))\|_2\big\}\,. \tag{54}$$

This is the empirical counterpart of the $\min\{\|x\|, \|x'\|\}$ factor appearing in the ReLU-NTK feature-map smoothness bound.

CNN instantiation. For a convolutional critic in the CNTK regime, we use the analogous local CNTK size factor. We do not require exact CNTK computation in the default algorithm. Let $\widetilde{m}(u, v)$ denote the local CNTK feature-map size factor for two embedded inputs u and v. Equivalently, $\widetilde{m}$ is any frozen upper-bound surrogate chosen before critic training and intended to satisfy a local CNTK smoothness inequality of the form

$$\|\psi_{\mathrm{CNTK}}(u) - \psi_{\mathrm{CNTK}}(v)\|_{\mathcal{H}} \le c_{\mathrm{cnn}}\sqrt{\widetilde{m}(u,v)\,\|u-v\|_2} + \widetilde{L}\|u-v\|_2. \tag{55}$$

The CNN geometry factor is then

$$\widehat{\mu}_J^{X,\mathrm{CNN}}(x) := \frac{1}{J}\sum_{\ell=1}^{J} \widetilde{m}\big(\zeta(x), \zeta(x_{(\ell)}(x))\big)\,. \tag{56}$$

Without an analytic verification of the CNTK smoothness condition, $\mathcal{U}_\Phi^{X,\mathrm{CNN}}$ is a heuristic frozen reliability score.

Unified state-action score. Let $J_\Phi := m_{\mathrm{loc}}$ be the fixed offline neighborhood size used to build the potential. For $\mathfrak{m} \in \{\mathrm{MLP}, \mathrm{CNN}\}$, define

$$\mathcal{U}_\Phi^{X,\mathfrak{m}}(x) := c_{1,\mathfrak{m}}\sqrt{\widehat{\mu}_{J_\Phi}^{X,\mathfrak{m}}(x)\,\bar{d}_{J_\Phi}^{X}(x)} + c_{2,\mathfrak{m}}\bar{d}_{J_\Phi}^{X}(x) + \sigma\sqrt{\frac{2\log(2N/\delta_U)}{J_\Phi}}. \tag{57}$$

Equation (5) suppresses the fixed embedding and model-family indices: $J = J_\Phi$, $U(x) = \mathcal{U}_\Phi^{X,\mathfrak{m}}(x)$, $\chi(x) = \widehat{\mu}_{J_\Phi}^{X,\mathfrak{m}}(x)$, and $\bar{d}(x) = \bar{d}_{J_\Phi}^{X}(x)$. Its constants are $C_1 = c_{1,\mathfrak{m}}$, $C_2 = c_{2,\mathfrak{m}}$, and $C_0 = \sigma\sqrt{2\log(2N/\delta_U)/J_\Phi}$. The default algorithm uses $\mathcal{U}_\Phi^{X,\mathfrak{m}}$ as a frozen reliability proxy. The constants $c_{1,\mathfrak{m}}$, $c_{2,\mathfrak{m}}$, and σ may come from analytic upper bounds or from an offline calibration split. If they are estimated, the fitted surrogate is frozen before the main critic training and does not use future critic predictions or gradients. Since J_Φ is fixed, the noise term is common across queries. Differences between scores come from distances and geometry factors. Figure 1 illustrates the local regression analysis. It does not validate the complete score as a return-error bound.

Canonical state projection. Let $z(s)$ be the state embedding used to build the graph. For each dataset state s_i, let $\mathcal{I}_k(s_i)$ be the index set of the k nearest offline states to s_i in the z-metric, including i. The state-side reliability score used by the potential is

$$\mathcal{U}_\Phi^{S,\mathfrak{m}}(s_i) := \max_{j\in\mathcal{I}_k(s_i)} \mathcal{U}_\Phi^{X,\mathfrak{m}}(s_j, a_j). \tag{58}$$

In Eq. (6), $\mathcal{N}_i = \mathcal{I}_k(s_i)$ and $U_i = \mathcal{U}_\Phi^{S,\mathfrak{m}}(s_i)$. This maximum is the canonical method definition. It is conservative and deterministic: a state is assigned the reliability of the least reliable nearby behavior action. High-quantile or behavior-action-weighted aggregation can be reported as ablations, but they are not part of the default algorithm.

Residual estimate. The theoretical bound in Corollary 2 contains the local residual $\mathcal{R}_M(x, r)$. The default reward construction does not use this term, because doing so would make the potential depend on the critic's online predictions. For reporting only, one may estimate

$$\widehat{\mathcal{R}}_M^{\mathrm{rep}}(x, r) := \max_{j:\, x_j\in\mathcal{B}_r(x)\cap\mathcal{C}_M} |Q_{\theta_M}(x_j) - y_j|, \tag{59}$$

where $\mathcal{C}_M$ is the set of distinct samples exposed after M mini-batch draws, and y_j is the fixed regression target used in the local analysis. This estimate is not fed back into Φ unless explicitly stated as an adaptive variant.

D.11 PRECOMPUTED GEOMETRY-CALIBRATED POTENTIAL

Definition 1 (Graph-smoothed return potential)**.** Let $\widehat{G}_i$ be standardized return-to-go labels at offline states, and let $W \in \mathbb{R}^{N\times N}$ be the symmetric graph-weight matrix built in the main text. Let

$$D_{\mathcal{G}} := \operatorname{diag}\left(\sum_{j=1}^{N} W_{ij}\right), \qquad L_{\mathcal{G}} := D_{\mathcal{G}} - W.$$

Return-to-go labels describe observed behavior continuations. We impose equality of fitted values at exactly identical states. Proximity in the embedding alone does not merge states. The unconstrained expressions below apply when all stored states are distinct. With duplicates, the exact reduced system given below implements the equality-constrained objective. Define

$$w_i := \frac{1}{(\mathcal{U}_{\Phi}^{S,\mathtt{m}}(s_i))^2 + \varepsilon}, \qquad D_{\Phi} := \operatorname{diag}(w_1, \ldots, w_N). \tag{60}$$

The dataset-state potential values, computed before critic training, solve

$$\phi := \arg\min_{\phi\in\mathbb{R}^N} \mathcal{L}_{\Phi}(\phi),$$

$$\mathcal{L}_{\Phi}(\phi) := \underbrace{\frac{1}{2}(\phi - \widehat{G})^{\top} D_{\Phi}(\phi - \widehat{G})}_{\text{weighted return fit}} + \underbrace{\frac{\tau}{2}\phi^{\top} L_{\mathcal{G}}\phi}_{\text{graph smoothing}}. \tag{61}$$

Lemma 3 (Unique bounded potential)**.** *If $\varepsilon > 0$ and $\tau \geq 0$, Eq.* (61) *has a unique minimizer. After clipping and convex k-NN interpolation, the resulting Φ is bounded by B_{Φ}.*

Proof. Because $\varepsilon > 0$, every confidence weight w_i is positive, so D_{Φ} is positive definite. Since $L_{\mathcal{G}}$ is positive semidefinite, $D_{\Phi} + \tau L_{\mathcal{G}}$ is positive definite. Differentiating Eq. (61) gives

$$\nabla_{\phi}\mathcal{L}_{\Phi}(\phi) = D_{\Phi}(\phi - \widehat{G}) + \tau L_{\mathcal{G}}\phi.$$

The unique stationary point is therefore

$$(D_{\Phi} + \tau L_{\mathcal{G}})\phi = D_{\Phi}\widehat{G}, \qquad \phi = (D_{\Phi} + \tau L_{\mathcal{G}})^{-1} D_{\Phi}\widehat{G}. \tag{62}$$

Clipping gives

$$\Phi(s_i) := \operatorname{clip}\big(\phi_i, -B_{\Phi}, B_{\Phi}\big). \tag{63}$$

For a novel state s, convex interpolation

$$\Phi(s) := \sum_{j\in\mathcal{N}_k(s)} \alpha_j(s)\Phi(s_j), \qquad \alpha_j(s) \geq 0, \qquad \sum_{j\in\mathcal{N}_k(s)} \alpha_j(s) = 1 \tag{64}$$

preserves the same bound. □

Lemma 4 (Balance of label fit and graph smoothing)**.** *For distinct graph nodes with positive weights, let $\deg_i = \sum_j W_{ij}$. At any node of positive degree, the unique solution satisfies*

$$\widehat{\phi}_i = \frac{w_i}{w_i + \tau\deg_i}\widehat{G}_i + \frac{\tau\deg_i}{w_i + \tau\deg_i}\sum_j \frac{W_{ij}}{\deg_i}\widehat{\phi}_j. \tag{65}$$

An isolated node retains its label. The identity follows by taking row i of Eq. (62) *and rearranging. It describes the estimator, not an ordering of its statistical risk.*

Exact reduction with repeated states. Let n_{Φ} be the number of distinct stored nonterminal states and let $R \in \{0,1\}^{N\times n_{\Phi}}$ map each occurrence to its state. Substituting $\phi = Rz$ into the occurrence-level objective gives

$$D_* = R^{\top} D_{\Phi} R, \quad L_* = R^{\top} L_{\mathcal{G}} R, \quad y_* = D_*^{-1} R^{\top} D_{\Phi}\widehat{G}, \quad z = (D_* + \tau L_*)^{-1} D_* y_*.$$

Every diagonal entry of D_* is positive. This is a weighted merge of labels and a summation of graph edges, not averaging independently fitted outputs. The per-node balance in Lemma 4 applies to this reduced system. The displayed occurrence-level balance assumes distinct states. Terminal successors are masked in shaping, not inserted as zero-pinned graph nodes. The statistical analysis uses the reduced node system. For any scalar $q > 0$, replacing (D_*, τ) by $(D_*/q, \tau/q)$ leaves the solution unchanged. Comparisons that normalize weights must therefore match the relative smoothing strength.

Remark 2 (Relation to density contrasts). A density contrast can highlight states that are frequent in high-return trajectories, but it need not account for local critic trainability. The graph-smoothed potential above instead uses return-to-go labels as task-oriented targets and modulates their influence by offline local-geometry reliability weights. The resulting potential is a surrogate for reward design.

Fixed-epoch safe transformation.

Proposition 6 (Safe reward transformation with epoch-dependent potential)**.** *For a fixed calibration epoch e, let*

$$\begin{aligned}\widetilde{r}_t^{(e)} &= \alpha_e r_t + \lambda_e\big(\gamma(1-d_t)\Phi(s_{t+1}) - \Phi(s_t)\big)\\ &= \alpha_e r_t + \gamma(1-d_t)F_e(s_{t+1}) - F_e(s_t),\\ F_e &:= \lambda_e\Phi.\end{aligned} \tag{66}$$

If $\alpha_e > 0$, $\lambda_e \geq 0$, and Φ is bounded, then under the terminal convention in Appendix D.7,

$$Q^{\pi}_{\widetilde{r}^{(e)}}(s,a) = \alpha_e Q^{\pi}_r(s,a) - F_e(s), \tag{67}$$

up to the admissible action-independent constant offset in the continuing case. In particular, optimal action ordering is preserved at every fixed epoch.

Proof. Since Φ is bounded by Lemma 3 and $\lambda_e \geq 0$, F_e is bounded. Apply the telescoping identity in Appendix D.7 with $F \rightsquigarrow F_e$. □

Corollary 3 (Epoch annealing preserves fixed-epoch action ordering)**.** *Let $e = 0, 1, 2, \ldots$, $\beta_e = \exp(-B\,e/N)$, $\lambda_e = \lambda_0\beta_e$, and $\alpha_e = 1 + (\alpha_0 - 1)\beta_e$, with $B, N > 0$, $\lambda_0 > 0$, and $\alpha_0 > 1$. Then $\beta_e > 0$, $\lambda_e > 0$, $\alpha_e > 1$, $\lambda_e \to 0$, and $\alpha_e \to 1$. For every fixed calibration epoch e, the shaped reward in Eq.* (66) *preserves optimal action ordering under the same terminal convention as Proposition 6.*

Proof. For every finite e, $\beta_e > 0$, so $\lambda_e > 0$. Since $\beta_e \in (0, 1]$ and $\alpha_0 > 1$, $\alpha_e = (1-\beta_e)\cdot 1 + \beta_e\alpha_0 > 1$. Also $\beta_e \to 0$ as $e \to \infty$, hence $\lambda_e \to 0$ and $\alpha_e \to 1$. Proposition 6 applies with $F_e = \lambda_e\Phi$. This is a fixed-epoch statement and does not imply path invariance for the epoch-varying neural training process. The wider hyperparameter range $\alpha_0 > 0, \lambda_0 \geq 0$ still satisfies the fixed-epoch safety condition when $\alpha_e > 0$ and F_e is bounded, but it is an ablation/generalization rather than the default amplifying curriculum. □

D.12 Compatibility across offline TD learners

Calibration replaces the observed reward in a TD target. This interface applies across behavior-regularized actor–critics, implicit value learners, and conservative critics, including ReBRAC, IDQL, and SCQ (Tarasov et al., 2023; Hansen-Estruch et al., 2023; Shimizu et al., 2024). Their policy-extraction rules and regularizers retain their original form. We describe the connections below, separating reward-interface compatibility from optimization invariance. The algebraic comparisons use fixed $\alpha > 0$ and $F = \lambda\Phi$, with corresponding exact value functions under Proposition 6's terminal convention.

TD3+BC and ReBRAC. TD3+BC combines a TD critic with a behavior-cloning penalty on the actor. ReBRAC refines this approach through implementation choices and decoupled actor and critic-side behavior regularization (Fujimoto & Gu, 2021; Tarasov et al., 2023). Calibration changes the reward summand while retaining target-action sampling, target-network updates, and regularization, including ReBRAC's target-action penalty. The resulting change in critic scale can alter its balance with behavior regularization, so interface compatibility does not require an invariant actor objective.

IQL. The corresponding values have the form

$$\widetilde{Q}(s,a) = \alpha Q(s,a) - F(s), \qquad \widetilde{V}(s) = \alpha V(s) - F(s).$$

At a fixed state, positive affine transformations commute with expectile regression: the signs of the residuals are unchanged and their squared losses are multiplied by α^2. Consequently, the corresponding advantage is

$$\widetilde{A}(s,a) = \widetilde{Q}(s,a) - \widetilde{V}(s) = \alpha\big[Q(s,a) - V(s)\big].$$

The state-only potential cancels. Pure shaping with $\alpha = 1$ therefore leaves these exact advantages unchanged. With reward scaling, however, a fixed inverse temperature ω gives actor weights $\exp(\omega\alpha A)$ rather than $\exp(\omega A)$, before any clipping. Thus the same IQL code can have a different effective actor weighting after calibration.

IDQL. IDQL generalizes implicit value learning and extracts actions from a diffusion behavior model using critic-derived weights or greedy candidate selection (Hansen-Estruch et al., 2023). Calibration enters its Q-function TD target. The diffusion behavior-cloning objective and policy-extraction rule retain their original form. For its expectile critic, the affine relation above applies to corresponding exact values. Positive scaling and state-only shifts preserve Q rankings on a fixed candidate set, while the expectile-based implicit weights depend only on the sign of $Q - V$. This extraction differs from IQL's exponential AWR rule: those weights are not the diffusion training objective in IDQL.

CQL. For the standard unit-temperature log-sum-exp conservative gap, write

$$\mathcal{C}_s(Q) = \log \int_{\mathcal{A}} \exp(Q(s,a))\,\mathrm{d}a - \mathbb{E}_{a\sim\mu(\cdot|s)}Q(s,a),$$

where $\mu(\cdot \mid s)$ is the behavior-action distribution. In a discrete action space the integral is a sum. A state-only shift cancels between the two terms:

$$\mathcal{C}_s(\alpha Q - F) = \mathcal{C}_s(\alpha Q).$$

For $\alpha = 1$, this gap is unchanged. For general α, the gap is not simply $\alpha\mathcal{C}_s(Q)$, because scaling also changes the action distribution inside the log-sum-exp. For a Bellman backup without an unscaled entropy term, corresponding transformed critics have squared residuals scaled by α^2. Even in that case, retaining the same conservative coefficient does not preserve the balance between the two losses. With a fixed entropy temperature, the Bellman residual need not follow this scaling either. Approximate fitting adds further differences.

SCQ. Strategically Conservative Q-Learning selectively penalizes actions identified as far from the data, using a conditional variational autoencoder to estimate this separation (Shimizu et al., 2024). Its critic retains a Bellman-fitting term, so the calibrated reward enters that term while the behavior model, OOD-action selection, and selective penalty retain their original form. SCQ's penalty differs from CQL's log-sum-exp gap above. The same algebra is not needed for this interface connection. Reward scaling can change the balance between Bellman fitting, selective conservatism, and the actor's entropy or behavior regularization.

Table 3 reports the matched empirical controls for TD3+BC, IQL, and CQL. ReBRAC, IDQL, and SCQ illustrate the wider reward-interface applicability.

D.13 Schedule design from effective exposure

The main text uses a fixed potential Φ and updates only scalar coefficients at calibration-epoch boundaries. Let $e = 0, 1, 2, \ldots$ denote the calibration-epoch index. The default effective-exposure schedule is

$$\beta_e := \exp\left(-\frac{B\,e}{N}\right), \qquad \lambda_e = \lambda_0\beta_e, \qquad \alpha_e = 1 + (\alpha_0 - 1)\beta_e, \tag{68}$$

where $\alpha_0 > 1$ and $\lambda_0 > 0$ specify the default nondegenerate curriculum. Define the per-epoch multiplier $q_{\mathrm{ep}} := \exp(-B/N)$. Then $\beta_e = q_{\mathrm{ep}}^e$ is the cumulative weight after e coefficient updates, and $\beta_{e+1} = q_{\mathrm{ep}}\beta_e$. The shaping strength is multiplied by q_{ep} each epoch, while the original reward scale anneals down to one. For $B = 256$ and $N = 10^6$, this gives $\beta_e = (\exp(-256/10^6))^e \approx 0.999744^e$. This schedule keeps the sampling scale B/N but advances it at epoch resolution rather than at every gradient step. The epoch clock is chosen for neural optimization because one sample exposure need not yield a stable critic gradient. The theoretical guarantee supplied by Corollary 3 is that the whole coefficient path remains in the fixed-epoch action-order-preserving class and asymptotically returns to the original reward labels. Performance-optimal decay is a separate question.

Relation to distinct coverage. Here M and e have different roles. The variable M is an occupancy counter: it denotes the cumulative number of actual mini-batch draws, equivalently gradient updates under one mini-batch per update. The variable e is an effective-exposure counter: it denotes how many calibration-epoch coefficient updates have been completed. If calibration epoch ℓ contains K_ℓ gradient updates, then the actual draw count before epoch e is $M_e := \sum_{\ell<e} K_\ell$. To make the distinction explicit, one can introduce an effective-update scale K_{eff}: K_{eff} raw TD updates count as one effective exposure unit, and $E_e := \sum_{\ell<e} K_\ell / K_{\text{eff}}$. The default choice $E_e = e$ corresponds to choosing calibration epochs whose lengths are measured in effective-update units.

For M mini-batch draws from the fixed offline dataset, the expected distinct-coverage factor is $p(M) = 1 - (1-\rho_B)^M$, with $\rho_B = B/N$ for uniform mini-batches without replacement. Hence the uncovered mass is

$$\begin{aligned} 1 - p(M) &= (1-\rho_B)^M = \exp(M \log(1-\rho_B)) \\ &= \exp\left(-\frac{BM}{N} + O\left(\frac{MB^2}{N^2}\right)\right) \approx \exp(-BM/N), \qquad B \ll N. \end{aligned} \tag{69}$$

This gives a sampling-motivated clock. In particular, after approximately one full-data pass of mini-batch draws, $M = N/B$, the exact uncovered-mass expression becomes $(1 - B/N)^{N/B} \approx \mathrm{e}^{-1}$. If one used the cumulative draw count M_e, the distinct-coverage clock would decay as $\exp(-BM_e/N)$. If one instead uses the effective-update count E_e, the clock becomes $\exp(-BE_e/N)$. The default Eq. (68) is the special case $E_e = e$: it treats a calibration epoch, not a single raw mini-batch update, as one effective exposure unit. This removes the offline prior more slowly than raw distinct-coverage counting.

A distinct-coverage variant replaces the default clock by the logged or expected coverage ratio

$$\begin{aligned} \widehat{p}_e &:= \frac{|\mathcal{C}_e|}{N} \quad \text{or} \quad p_e := p(M_e), \\ \beta_e^{\text{cov}} &:= 1 - \widehat{p}_e \ \text{ or } \ 1 - p_e, \\ \lambda_e &= \lambda_0 \beta_e^{\text{cov}}, \qquad \alpha_e = 1 + (\alpha_0 - 1)\beta_e^{\text{cov}}. \end{aligned} \tag{70}$$

This variant still leaves Φ fixed. It only changes the scalar multiplier according to observed or expected mini-batch exposure over the fixed offline dataset.

D.14 Additional implementation details for reward calibration

Return labels. For a transition i at trajectory time t_i, the return-to-go label is

$$G_i := \sum_{\ell \ge t_i} \gamma^{\ell - t_i} r_\ell,$$

with the sum stopped at the terminal transition of the same trajectory. The standardized label can be implemented as

$$\widehat{G}_i := \mathrm{clip}\left(\frac{G_i - \mu_G}{\sigma_G + \varepsilon_G}, -B_G, B_G\right),$$

where μ_G, σ_G are computed on a declared fitting or calibration set and frozen. True terminal states have zero continuation potential. Time limits and cut dataset trajectories require separate treatment. A truncated return without a valid tail estimate contributes label bias, accounted for in Appendix D.1. The analytical example in Appendix D.2 uses complete continuations and identity preprocessing to isolate graph smoothing and label weighting.

From return fitting to shaping labels. For $\varepsilon_G > 0$, the standardization above has affine coefficients $a_G = (\sigma_G + \varepsilon_G)^{-1}$ and $b_G = -\mu_G/(\sigma_G + \varepsilon_G)$ before clipping. The following identity also allows arbitrary fixed a_G, b_G.

Lemma 5 (Shaping labels relative to trajectory returns)**.** *Consider a finite trajectory with $d_t = 0$ for $t < T-1$ and $d_{T-1} = 1$, and define $G_T = 0$ and $G_t = r_t + \gamma(1-d_t)G_{t+1}$. For fixed scalars a_G, b_G and a state function Φ, let $\epsilon_t^\Phi = \Phi(s_t) - (a_G G_t + b_G)$ for $0 \le t \le T$. On every transition $0 \le t < T$,*

$$h_t = -a_G r_t - b_G\big[1 - \gamma(1-d_t)\big] + \gamma(1-d_t)\epsilon_{t+1}^\Phi - \epsilon_t^\Phi. \tag{71}$$

For $a_G = 1$, $b_G = 0$, and $\Phi(s_t) = G_t$ at the nonterminal states, $h_t = -r_t$ and $\widetilde{r}_t^{(e)} = (\alpha_e - \lambda_e) r_t$ along this trajectory.

Proof. Substitute $\Phi(s_t) = a_G G_t + b_G + \epsilon_t^\Phi$ in the definition of h_t:

$$h_t = a_G\big[\gamma(1-d_t)G_{t+1} - G_t\big] + b_G\big[\gamma(1-d_t) - 1\big] + \gamma(1-d_t)\epsilon_{t+1}^\Phi - \epsilon_t^\Phi.$$

The return recursion makes the first bracket $-r_t$, which gives the identity. In the raw-return special case, the discrepancies vanish at $t < T$, and the terminal mask removes the final next-state term. Substitution in Eq. (3) gives the calibrated reward. □

The discrepancy is measured from the untruncated affine label, so it includes the effects of label clipping, shared graph fitting, final potential clipping, and the chosen interpolation. Repeated states may have different trajectory returns. Each occurrence then has its own ϵ_t^Φ, while the constructed Φ remains a single state function.

Graph and solver. The graph is built on normalized states, PCA features, or frozen encoder features before critic training. For large N, W should be a sparse k-NN graph. The linear system in Eq. (62) can be solved by conjugate gradient or preconditioned conjugate gradient, with each iteration costing $O(N + |E_{\mathcal{G}}|)$ for graph edge set $E_{\mathcal{G}}$. Landmark or Nyström approximations are natural alternatives when N is very large.

Schedule and coverage logging. The default schedule uses the calibration-epoch clock in Eq. (68). For ablations, one may log theoretical p_e or a realized distinct-coverage estimate $\widehat{p}_e = |\mathcal{C}_e|/N$. Exact counters are simple for moderate offline datasets, and approximate distinct counters are suitable for large datasets. The default method does not recompute Φ during critic training. Logging affects only the scalar schedule.

Fallbacks and hyperparameters. If $M = 0$ or $J_M^{\mathrm{raw}}(m_{\mathrm{loc}}, \eta) > N$, the target-effective-size condition is inactive for that point in training. Training still uses the fixed potential and the epoch-paced scalar schedule. The principal settings are the neighborhood counts, graph-kernel parameters, smoothing strength τ, weight floor ε, clipping thresholds B_Φ, B_G, initial coefficients α_0, λ_0, and epoch lengths K_e, with $\alpha_0 > 1$ and $\lambda_0 > 0$ for the default schedule. They should be fixed across a benchmark family when possible, or selected on a validation split without using evaluation returns.

Remark 3 (Connection to reward calibration). The construction can be read as a three-step design: the offline MLP/CNN local reliability score controls how strongly return labels enter the fixed graph potential, the calibration-epoch clock controls how quickly the shaping coefficient is removed, and each fixed epoch remains an ordinary potential-shaped reward with $F_e = \lambda_e \Phi$. The geometry score motivates the weights, but its calibration to return uncertainty is an additional condition, not a consequence of the local critic bound. Appendix D.1 separates fixed-stage policy preservation from conditional comparisons of potential quality.

E Reward-Drift Component: Algorithm and Proofs

E.1 Reward-Drift-Aware Relabeling and Rejuvenation

This appendix gives the reward-drift-aware relabeling and rejuvenation (RDRR) component summarized in Section 4.3. It applies when transition dynamics remain fixed and reward labels can be recomputed, or approximated, on stored transitions $z = (s, a, r, s', d)$. Here $r_t(z)$ denotes the phase-indexed label whose drift is tracked, and r is the stored original reward. In the calibration curriculum, $r_t(z) = \widetilde{r}_{e(t)}(z)$, where $e(t)$ is the calibration epoch active at phase t. RDRR complements fixed-epoch policy preservation by addressing critic-side tracking as these labels change.

Design principle. Reward drift controls reset intensity, dormancy controls local neuron recycling, and reward sensitivity controls which parameter groups are reset.

Unified intervention rule. Let Q_θ be a neural action-value function and partition its parameters into groups $\theta = (\theta_g)_{g\in\mathcal{G}}$, where g may denote a head, a hidden layer, or an individual neuron. We use a single rejuvenation operator to cover both reset and Recycling Dormant Neurons (ReDo) (Sokar et al., 2023):

$$\mathcal{R}_{m,\alpha}(\theta)_g = \big(1 - m_g(t)\alpha_g(t)\big)\theta_g + m_g(t)\alpha_g(t)\widetilde{\theta}_g, \qquad \widetilde{\theta}_g \sim \mathcal{I}_g, \tag{72}$$

where $\mathcal{I}_g$ is the group-specific reset distribution, such as the original layer initialization distribution or a calibrated head-recovery prior. The activity score is

$$u_{\ell i}(t) = \frac{\mathbb{E}_{x\sim\mathcal{B}_t}[|h_i^\ell(x)|]}{\frac{1}{n_\ell}\sum_{j=1}^{n_\ell}\mathbb{E}_{x\sim\mathcal{B}_t}[|h_j^\ell(x)|] + \varepsilon}, \tag{73}$$

and ReDo corresponds to the fine-grained case where dormant neurons are reinitialized and their outgoing weights are zeroed. To decide when stronger reset is needed, we estimate reward drift on an anchor set $\mathcal{A}_t$. Unless stated otherwise, $\mathcal{A}_t$ is a fixed stratified subset of n_A stored transitions drawn at the beginning of the current reward phase, excluded from gradient updates, and refreshed only when the reward phase changes. Here Δ is a chosen lookback window, b_t is a drift baseline, and $c_r > 0$ is a drift scale:

$$\widehat{\Delta}_t^r = \left(\frac{1}{|\mathcal{A}_t|}\sum_{z\in\mathcal{A}_t}\big(r_t(z) - r_{t-\Delta}(z)\big)^2\right)^{1/2}, \tag{74}$$

$$\lambda_t = \mathrm{clip}\left(\frac{\widehat{\Delta}_t^r - b_t}{c_r}, 0, 1\right), \tag{75}$$

so small λ_t recovers ReDo-like behavior and large λ_t favors stronger reward-sensitive reset. The empirical RMS drift $\widehat{\Delta}_t^r$ is a practical intervention proxy. The tracking theorem below uses the sup-norm drift $\Delta_t^r = \|r_t - r_{t-1}\|_\infty$. When connecting the algorithmic proxy to the theorem, one needs an anchor-set coverage or high-quantile assumption that upper-bounds the relevant reward drift on the domain being tracked.

For each parameter group g, define the reward-sensitivity score

$$\begin{aligned}\rho_g(t) &= \left\|\mathbb{E}_{z\sim\mathcal{B}_t}\left[\Delta r_t(z)\nabla_{\theta_g}Q_\theta(s,a)\right]\right\|_2, \\ \Delta r_t(z) &= r_t(z) - r_{t-\Delta}(z),\end{aligned} \tag{76}$$

and the importance score

$$I_g(t) = \mathbb{E}_{(s,a)\sim\mathcal{B}_t}\left[\left\|\nabla_{\theta_g}Q_\theta(s,a)\right\|_2^2\right]. \tag{77}$$

For a general group g, let $u_g(t)$ be the mean activity score of its constituent neurons, using Eq. (73) for singleton neuron groups. Let $\mathcal{G}_{\ell(g)}$ be the groups in the same layer or module as g, and define the raw dormancy gap $d_g^{\mathrm{raw}}(t) := (\tau_d - u_g(t))_+$. With a denominator floor $\varepsilon_{\mathrm{norm}} > 0$, the default normalized scores are

$$\begin{aligned}\bar{\rho}_g(t) &:= \frac{\rho_g(t)}{|\mathcal{G}_{\ell(g)}|^{-1}\sum_{g'\in\mathcal{G}_{\ell(g)}}\rho_{g'}(t) + \varepsilon_{\mathrm{norm}}}, \\ \bar{d}_g(t) &:= \frac{d_g^{\mathrm{raw}}(t)}{|\mathcal{G}_{\ell(g)}|^{-1}\sum_{g'\in\mathcal{G}_{\ell(g)}}d_{g'}^{\mathrm{raw}}(t) + \varepsilon_{\mathrm{norm}}}, \\ \bar{I}_g(t) &:= \frac{I_g(t)}{|\mathcal{G}_{\ell(g)}|^{-1}\sum_{g'\in\mathcal{G}_{\ell(g)}}I_{g'}(t) + \varepsilon_{\mathrm{norm}}}.\end{aligned} \tag{78}$$

The constants $\mu, c_\rho, c_d \geq 0$ control the importance penalty and the reward-sensitivity and dormancy contributions, and η_t is either a fixed threshold or a quantile threshold controlling the intervention budget. We then score and actuate interventions by

$$S_g(t) = \lambda_t\bar{\rho}_g(t) + (1-\lambda_t)\bar{d}_g(t) - \mu\bar{I}_g(t), \tag{79}$$

$$\begin{aligned}\alpha_g(t) &= \mathrm{clip}\big(a_{\mathrm{reset}} + c_\rho\lambda_t\bar{\rho}_g(t) + c_d(1-\lambda_t)\bar{d}_g(t),\, 0, 1\big), \\ m_g(t) &= \mathbf{1}\{S_g(t) \geq \eta_t\},\end{aligned} \tag{80}$$

where $a_{\mathrm{reset}} \in [0,1]$ is the baseline reset intensity, distinct from the initial reward scale $\alpha_0 > 1$. The control rule interpolates between dormant-neuron recycling and reward-sensitive reset.

Current-reward relabeling. Because the dynamics and stored transitions are fixed, offline data become stale mainly through their reward labels rather than through transitions. The dataset should therefore retain enough context to recompute current rewards whenever possible, and the TD target is relabeled as in Eq. (11). When exact relabeling is unavailable, we fall back to recency weighting or lightweight head recovery. Implementation details are given in Appendix E.6. In practice encoder parameters are protected more strongly than heads or late layers, since dynamics-relevant features remain reusable across reward changes.

Procedure. Algorithm 2 interleaves ordinary current-reward TD fitting with occasional intervention rounds. Most updates simply relabel the current TD target and train the base learner. Every C steps, the method estimates reward drift, scores parameter groups by reward sensitivity, dormancy, and importance, and then applies rejuvenation only where the drift-conditioned score is large enough. The default implementation keeps $\mathcal{A}_t$ fixed within a reward phase so that changes in $\widehat{\Delta}_t^r$ reflect reward drift rather than anchor-resampling noise.

Algorithm 2 Reward-drift-aware relabeling and rejuvenation (RDRR)

Data Offline dataset $\mathcal{D}$ and phase anchor set $\mathcal{A}_t$ with n_A transitions.
Model Critic Q_θ, target critic $Q_{\bar{\theta}}$, groups $\mathcal{G}$, and reset priors $\{\mathcal{I}_g\}_{g\in\mathcal{G}}$.
Timing Intervention interval C and lookback window Δ.
Scores Drift parameters b_t, c_r, dormancy threshold τ_d, and floors $\varepsilon, \varepsilon_{\text{norm}}$.
Control Reset baseline a_{reset}, penalty μ, weights c_ρ, c_d, and threshold η_t.

for training step $t = 1, 2, \ldots$ **do**
  Relabel the target using Eq. (11) and update the base learner
  if t is an intervention step (every C steps) **then**
    Estimate $\widehat{\Delta}_t^r$ and λ_t using Eqs. (74) and (75)
    Compute $S_g(t)$, $m_g(t)$, and $\alpha_g(t)$ using Eqs. (76)–(80)
    Apply $\mathcal{R}_{m,\alpha}$ and reset the affected optimizer states
    if a value head was reset **then**
      Run head recovery
    end if
  end if
end for

E.2 Bellman Drift Induced by Reward Drift

Lemma 6 (Bellman drift)**.** *For any $t \geq 2$,*

$$\left\|Q_t^* - Q_{t-1}^*\right\|_\infty \leq \frac{\|r_t - r_{t-1}\|_\infty}{1-\gamma} = \frac{\Delta_t^r}{1-\gamma}. \tag{81}$$

Writing $V_T^r = \sum_{t=2}^T \Delta_t^r$ for the cumulative reward variation gives

$$\sum_{t=2}^{T} \left\|Q_t^* - Q_{t-1}^*\right\|_\infty \leq \frac{V_T^r}{1-\gamma}. \tag{82}$$

Proof. Since $\mathcal{T}_t$ and $\mathcal{T}_{t-1}$ are γ-contractions in the sup norm,

$$\begin{aligned}
\left\|Q_t^* - Q_{t-1}^*\right\|_\infty &= \left\|\mathcal{T}_t Q_t^* - \mathcal{T}_{t-1} Q_{t-1}^*\right\|_\infty && (83)\\
&\leq \left\|\mathcal{T}_t Q_t^* - \mathcal{T}_t Q_{t-1}^*\right\|_\infty + \left\|\mathcal{T}_t Q_{t-1}^* - \mathcal{T}_{t-1} Q_{t-1}^*\right\|_\infty && (84)\\
&\leq \gamma \left\|Q_t^* - Q_{t-1}^*\right\|_\infty + \|r_t - r_{t-1}\|_\infty . && (85)
\end{aligned}$$

Rearranging gives the first inequality. Summing over t gives the second. □

E.3 Perturbation Bound for ReDo

Assumption 2. For any subset S of already recycled candidate neurons and any remaining candidate neuron (ℓ, i) in the selected set, the partially intervened network obtained after resetting S is locally

Lipschitz on the batch domain $\mathcal{B}_t$ with constant $C_{\ell i}$ with respect to the activation h_i^ℓ. For the $O(\tau_d)$ specialization below, assume additionally that selected neurons satisfy a batch max-to-mean control $\sup_{x\in\mathcal{B}_t} |h_i^\ell(x)| \le C_u \mathbb{E}_{x\sim\mathcal{B}_t}[|h_i^\ell(x)|]$ with a finite constant C_u.

Proposition 7. *Let $\mathcal{D}_t = \{(\ell, i) : u_{\ell i}(t) \le \tau_d\}$ be the set of ReDo-selected neurons. If ReDo resets their incoming weights and sets their outgoing weights to zero, then*

$$\|Q_{\theta^+} - Q_\theta\|_{\mathcal{B}_t,\infty} \le \sum_{(\ell,i)\in\mathcal{D}_t} C_{\ell i} \sup_{x\in\mathcal{B}_t} |h_i^\ell(x)|. \tag{86}$$

In particular, if the normalized activity scores satisfy $u_{\ell i}(t) \le \tau_d$, the layer-wise mean activity is bounded, and the batch max-to-mean control in Assumption 2 holds, then

$$\|Q_{\theta^+} - Q_\theta\|_{\mathcal{B}_t,\infty} = O(\tau_d).$$

Proof. For a feed-forward critic, order the recycled neurons from downstream layers to upstream layers and reset them sequentially, with outgoing-weight zeroing taking precedence on shared connections. Previously reset downstream neurons then leave each remaining neuron's activation unchanged. For a single neuron (ℓ, i), setting the outgoing weight $W_{:,i}^{\ell+1}$ to zero removes its immediate contribution to downstream activations. Under Assumption 2, after any partial intervention the resulting incremental change in the network output on $\mathcal{B}_t$ is at most $C_{\ell i} \sup_{x\in\mathcal{B}_t} |h_i^\ell(x)|$. A telescoping sum over the sequential reset path therefore gives (86). The $O(\tau_d)$ statement follows from the definition of normalized activity in (73), bounded layer-wise mean activity, and the max-to-mean control. □

E.4 Conditional Tracking Statement for RDRR

Under the approximate-contraction condition below, relabeled Bellman fitting with selective rejuvenation has a tracking-error bound controlled by the reward-variation budget.

Assumption 3. Let $e_t = \|Q_{\theta_t} - Q_t^*\|_\infty$. After one training phase under reward r_t, including relabeled Bellman fitting and the RDRR rejuvenation step, suppose that for $t = 1, \dots, T-1$,

$$e_{t+1} \le \beta e_t + \frac{\Delta_{t+1}^r}{1-\gamma} + \varepsilon_t, \qquad 0 < \beta < 1, \tag{87}$$

where $\varepsilon_t = \varepsilon_{\mathrm{stat},t} + \varepsilon_{\mathrm{opt},t} + \varepsilon_{\mathrm{approx},t} + \varepsilon_{\mathrm{reset},t}$.

Theorem 4 (Tracking under an approximate-contraction condition)**.** *Under Assumption 3,*

$$e_T \le \beta^{T-1} e_1 + \sum_{t=1}^{T-1} \beta^{T-1-t} \left(\frac{\Delta_{t+1}^r}{1-\gamma} + \varepsilon_t \right), \tag{88}$$

$$\sum_{t=1}^{T} e_t \le \frac{e_1}{1-\beta} + \frac{V_T^r}{(1-\beta)(1-\gamma)} + \frac{\sum_{t=1}^{T-1} \varepsilon_t}{1-\beta}. \tag{89}$$

Proof. Starting from Assumption 3, for $t = 1, \dots, T-1$,

$$e_{t+1} \le \beta e_t + a_t, \qquad a_t = \frac{\Delta_{t+1}^r}{1-\gamma} + \varepsilon_t. \tag{90}$$

Unrolling the recursion gives

$$e_T \le \beta^{T-1} e_1 + \sum_{t=1}^{T-1} \beta^{T-1-t} a_t, \tag{91}$$

which proves (88). For the cumulative bound,

$$\sum_{t=1}^{T} e_t \le \sum_{t=1}^{T} \beta^{t-1} e_1 + \sum_{t=2}^{T} \sum_{k=1}^{t-1} \beta^{t-1-k} a_k \tag{92}$$

$$\le \frac{e_1}{1-\beta} + \frac{1}{1-\beta} \sum_{k=1}^{T-1} a_k \tag{93}$$

$$= \frac{e_1}{1-\beta} + \frac{1}{1-\beta} \left(\sum_{k=1}^{T-1} \frac{\Delta_{k+1}^r}{1-\gamma} + \sum_{k=1}^{T-1} \varepsilon_k \right) \tag{94}$$

$$= \frac{e_1}{1-\beta} + \frac{V_T^r}{(1-\beta)(1-\gamma)} + \frac{\sum_{k=1}^{T-1} \varepsilon_k}{1-\beta}, \tag{95}$$

which proves (89). □

E.5 Idealized dynamic-regret corollary

If the deployed policy π_t is greedy with respect to Q_{θ_t}, define the dynamic regret

$$\mathrm{DReg}_T = \sum_{t=1}^{T} \left[V_t^*(s_0) - V_t^{\pi_t}(s_0) \right].$$

For exact-greedy control, the standard performance-loss bound uses the sup-norm approximation error over the relevant state-action domain. Extending it to regularized offline actor updates such as TD3+BC, IQL, or CQL requires additional policy-constraint and support residual terms. In the exact-greedy setting,

$$\mathrm{DReg}_T \le \frac{2e_1}{(1-\gamma)(1-\beta)} + \frac{2V_T^r}{(1-\gamma)^2(1-\beta)} + \frac{2\sum_{t=1}^{T-1} \varepsilon_t}{(1-\gamma)(1-\beta)}. \tag{96}$$

Proof. The standard approximate-greedy performance-loss inequality gives

$$V_t^*(s) - V_t^{\pi_t}(s) \le \frac{2}{1-\gamma} \left\| Q_{\theta_t} - Q_t^* \right\|_\infty = \frac{2}{1-\gamma} e_t.$$

Summing over t and substituting Eq. (89) yields Eq. (96). □

E.6 Additional Implementation Details

Fallback when relabeling is unavailable. If the current reward cannot be recomputed exactly, a weaker substitute is recency-weighted TD learning:

$$\mathcal{L}_t(\theta) = \mathbb{E}_{(s_i, a_i, r_i, s_i', d_i) \sim \mathcal{D}} \left[w_{t,i} \left(r_i + \gamma(1-d_i) V_{\mathrm{boot}}(s_i') - Q_\theta(s_i, a_i) \right)^2 \right],$$

$$w_{t,i} \propto \exp\left(-\frac{t - \tau_i}{\kappa} \right),$$

where τ_i is the insertion time or reward-label timestamp of transition i. As in Eq. (11), V_{boot} is the base learner's bootstrap value.

Head recovery after reset. For reward-only drift, a useful architecture is $Q_\theta(s,a) = w^\top \phi_\psi(s,a)$, where ϕ_ψ is a representation network and w is a value head. At a fixed recovery phase t, define

$$Y_i := r_t(z_i) + \gamma(1-d_i) V_{\mathrm{boot}}(s_i'), \qquad z_i = (s_i, a_i, r_i, s_i', d_i).$$

The target and bootstrap parameters are held fixed during recovery. Under the calibration curriculum, $Y_i = Y_{e(t)}(z_i)$ from Eq. (11). After resetting the head, one can reduce collapse by solving

$$w_t^+ = \arg\min_w \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(w^\top \phi_\psi(s_i, a_i) - Y_i \right)^2 + \lambda_w \left\| w - \widetilde{w} \right\|_2^2 \right\}, \tag{97}$$

$$\widetilde{w} \sim \mathcal{I}_w.$$

Optimizer state. Whenever a parameter group is intervened on, the corresponding adaptive optimizer states should be reset:

$$m_g \leftarrow 0, \qquad v_g \leftarrow 0.$$

Otherwise stale momentum or second-moment estimates can immediately undo the intended reset.

Choosing groups. The group set $\mathcal{G}$ may include multiple granularities, but interventions should be thresholded within a fixed granularity or module pool whenever possible. A practical choice is to maintain separate pools such as

$$\mathcal{G} = \left\{ \begin{array}{l} \text{critic head},\ \text{actor head}, \\ \text{last hidden layer},\ \text{individual hidden neurons} \end{array} \right\}.$$

The encoder can be included with a high threshold or strong importance penalty, so it is reset only under severe reward drift or representation collapse. If mixed granularity is retained in one shared pool, it should be accompanied by explicit per-granularity normalization or separate quantile thresholds.

Normalizing reward sensitivity. The normalized score $\bar{\rho}_g(t)$ can be obtained by layer-wise or per-granularity normalization,

$$\bar{\rho}_g(t) = \frac{\rho_g(t)}{\frac{1}{|\mathcal{G}_\ell|} \sum_{g' \in \mathcal{G}_\ell} \rho_{g'}(t) + \varepsilon_{\text{norm}}},$$

where $\mathcal{G}_\ell$ is the set of groups within the same layer or module. This avoids always selecting large modules merely because they contain more parameters and is especially important when group pools mix heads, layers, and individual neurons.

Intervention thresholds. The threshold η_t can be fixed or chosen as a quantile of $\{S_g(t) : g \in \mathcal{G}\}$. A quantile threshold gives direct control over the intervention budget per rejuvenation step.

AI USE STATEMENT

Generative AI tools assisted with language revision, organization of the technical presentation, and iterative discussion of the reward-calibration formulation and analysis. The authors reviewed the resulting material and remain responsible for the mathematical arguments, experimental results, citations, and conclusions.