CRITIC-GROUNDED ONLINE FINE-TUNING WITH ACTIVE PREFERENCE ELICITATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The offline-to-online paradigm in reinforcement learning deploys a pretrained actor-critic checkpoint without access to the offline dataset or the reward function that produced it. The data may be withheld for privacy or proprietary reasons, and the reward may have been handcrafted by experts. We study online fine-tuning in this regime, where the only available supervision is preference feedback: a labeler compares two short behavior segments, may abstain, and never provides a per-step reward. Two obstacles arise in this setting. First, a reward learned from preferences is identified only up to a scale and an offset, which the standard Bradley-Terry likelihood cannot pin down, whereas the pretrained critic holds values in the units of the offline reward. Writing an arbitrarily placed reward into the replay buffer corrupts those values and destabilizes fine-tuning. We design a critic gauge, which recovers the reward implied by the frozen critic and regresses the learned reward onto it to fix the scale and offset. Second, preference labels are costly, and the labeler may abstain, so comparisons must be chosen to be both informative and answerable. We score each candidate by the expected reduction in posterior variance across the pool, under a Gaussian posterior over the reward model's parameters, weighted by a calibrated online estimate of the labeler's likelihood of answering. Our selection is sequential, folding each label into the posterior before the next pair is chosen so that redundant comparisons are never selected. On eight standard RL environments, our method closely matches WSRL, a baseline that shares our no-data-retention setting but observes the true online reward, and substantially outperforms PEBBLE, which learns from preferences like ours but starts without a pretrained checkpoint.

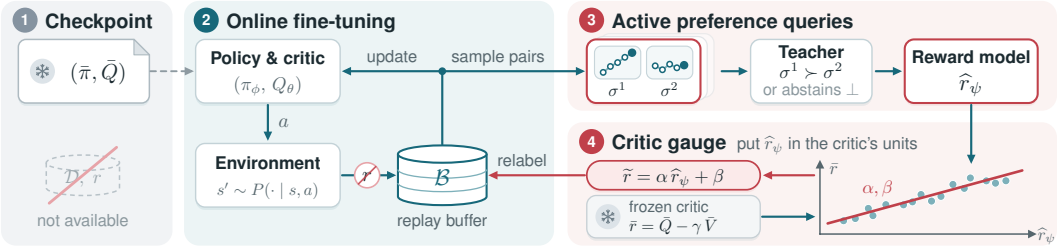
1 INTRODUCTION

In modern machine learning, models are commonly pretrained on diverse data and then fine-tuned for a specific task (Devlin et al., 2019; Driess et al., 2023; Radford et al., 2021; Zhai et al., 2023; Touvron et al., 2023; Zhou et al., 2025a). Reinforcement learning follows a similar paradigm: a first phase trains on offline data without interaction with the environment, and a second phase fine-tunes online. The central difficulty is stability, that is, keeping the online phase from forgetting what offline training produced, and it is usually handled by retaining the offline data and co-training on it. But the offline data is often unavailable. It may be withheld for privacy, proprietary, or redistribution reasons, so that a downstream user receives a checkpoint but not the dataset from which it was derived. Even when it is available, carrying a large dataset alongside the checkpoint is a storage burden. This motivates fine-tuning directly from the pretrained checkpoint. WSRL (Zhou et al., 2025b) addresses this by seeding online RL with a few warm-up roll-outs from the frozen pretrained actor, thereby allowing offline data to be discarded without destabilizing fine-tuning.

The reward model used to train the offline checkpoint may also be unavailable. The entity that fine-tunes the policy online is often not the one that did offline pretraining, and the reward used offline may have been assembled from costly sensors and hand-crafted by domain experts. For instance, consider a locomotion policy shipped to an operator who wants to adapt it to their own conditions. They can watch the agent act and say which of two short clips looks better, but they cannot read out a scalar reward at every timestep, and the sensor that produced the training reward stays with the vendor. The natural signal in this situation is preference feedback: comparisons between behavior segments, from which a reward model is learned online (Christiano et al., 2017). PEBBLE (Lee et al., 2021) learns such a reward, but it begins from scratch, with no offline or task-specific pretraining, only

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.

unsupervised exploration and no critic, and so forgoes the offline prior that makes fine-tuning sample-efficient. Motivated by these, we study online fine-tuning from an offline checkpoint, with preferences as the only supervision. Two obstacles arise here that neither WSRL (which sees the true online reward) nor PEBBLE (which has no offline prior) confronts simultaneously.



Placing the learned reward: the critic gauge. A reward model learned from preferences is not directly usable in the fine-tuning loop. The Bradley-Terry likelihood (Bradley & Terry, 1952), typically used to learn a reward model, is invariant to a constant shift of the reward and cannot pin down its scale, so preferences identify the reward only up to an unknown scale and offset. The critic, however, is not invariant to either: the pretrained critic holds values in the units of the true reward, and our online fine-tuner (an off-policy SAC agent) mixes the learned reward with those values in its Bellman target. Writing an arbitrarily placed reward into the replay buffer, therefore, corrupts the very estimates the offline phase produced, before the policy can improve. The usual fix rescales the learned reward to the statistics of the true reward, assuming access to the offline data. Our first contribution, the *critic gauge*, needs no such access. The frozen critic is approximately Bellman-consistent with respect to the unknown true reward and can be read directly from the checkpoint at one evaluation per transition, after a few warm-up roll-outs. Regressing the preference-learned reward onto it recovers the scale and offset, placing the preference reward in the critic's own units.

Spending a small budget: active selection. Preference labels (comparisons between behavior segments) are expensive to collect because expert time is scarce, and automated evaluators are costly. Moreover, a labeler may simply abstain from comparing two indistinguishable segments, spending a query budget for nothing. Hence, only a small number of comparisons can be asked. So, the learner must choose them to be both informative and answerable, since the most uncertain pairs are frequently the ones the labeler cannot order. Our second contribution is an *active selection rule*. Maintaining a Gaussian posterior over the reward model's weights, it first scores each candidate pair of segments by how much a label for it would shrink the posterior uncertainty over the weights across the whole pool of segments. Then it weights the uncertainty scores using a calibrated online estimate of the labeler's likelihood of answering and selects the pair with the highest score for querying. Selection is sequential and greedy: each query is answered and folded into the posterior before the next pair is chosen, so a pair is never selected without accounting for what the already-labeled pairs have resolved, which suppresses redundant queries under a tight budget.

Contributions and results. We propose (i) the *critic gauge*, which calibrates the scale and offset of a preference-learned reward against the frozen critic, without the offline dataset or the original reward function; (ii) an *active selection* rule that scores comparisons by expected posterior-variance reduction across the pool and reweights them by labeler's answerability, so that queries are both informative and likely to be answered; (iii) an empirical study on eight RL domains including Adroit and locomotion tasks. We position our algorithm, ANCOR, between two prior methods: (a) WSRL (Zhou et al., 2025b) that starts with a checkpoint with no retained offline data, but observes the true online reward, and so serves as a skyline that measures the cost of not seeing the reward. Our method closely matches it across several environments despite learning the reward solely from preferences under a tight budget, showing the importance of our *critic gauge* and *active selection rule*; (b) PEBBLE (Lee et al., 2021) which learns the reward from preference supervision but starts without an offline checkpoint, and so serves as a baseline. We outperform it by a wide margin, showing the importance of offline prior, once its reward is correctly placed using our *critic gauge*.

1.1 Related Work

Offline-to-online RL fine-tunes an offline-pretrained actor-critic through online interaction, and doing so naively with off-policy RL often causes an immediate drop in performance (Nair et al., 2020; Lee et al., 2022; Nakamoto et al., 2023; Zhang et al., 2024). Prior work restores stability in two

ways. On the critic side, methods perturb value updates (Zhang et al., 2024), calibrate conservative values (Nakamoto et al., 2023), or add an online pre-training phase (Shin et al., 2025). On the replay and constraint side, they rebalance or mix offline data into online updates (Lee et al., 2022; Ball et al., 2023), adapt or relax policy constraints (Wang et al., 2023; Li et al., 2023b), or chunk actions for coherent exploration (Li et al., 2025b). Li et al. (2025a) frame these choices as a stability-plasticity trade-off. Most of these methods retain the offline data during fine-tuning (Zhou et al., 2025b), and all of them receive the same reward offline and online. When the reward is unavailable, the natural supervision is preference feedback Christiano et al. (2017). Active query selection for preference-based reward learning has been studied by Sadigh et al. (2017) and Biyik et al. (2020), the latter favoring queries that the human can answer easily. Few works that combine preferences with pretraining stop short of our setting. Agnihotri et al. (2024) warm-start from offline preferences in bandits, Macuglia et al. (2026) fine-tune a policy pretrained by behavioral cloning, with no offline critic and Hejna & Sadigh (2023) pretrain the reward model rather than the actor-critic. Methods that correct a mismatched reward, through residual rewards (Cao et al., 2025), residual Q-learning (Li et al., 2023a), or reward transfer (Zhu et al., 2023), assume that part of the reward is known, either as a prior or as an online signal. We study fine-tuning from a single checkpoint, with preferences as the only supervision, in which the learned reward must be expressed in the critic's units.

2 Problem Setting

Consider a Markov decision process with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $P(\cdot \mid s, a)$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ gives a probability distribution over actions for each state and induces the expected discounted return $J(\pi) = \mathbb{E}_\pi\big[\sum_t \gamma^t r(s_t, a_t)\big]$. The Q-function of π is $Q^\pi(s, a) = \mathbb{E}_\pi\big[\sum_t \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\big]$ and $V^\pi(s) = \mathbb{E}_\pi\big[Q^\pi(s, a)\big]$ denotes the state-value.

An actor $\overline{\pi}$ and a critic $\overline{Q}$ are pretrained on an offline dataset $\mathcal{D} = \{(s, a, r, s')\}$. The online agent receives the checkpoint $(\overline{\pi}, \overline{Q})$. But it doesn't have access to the dataset $\mathcal{D}$, which is typically the case when a policy is shipped. The agent interacts with the MDP for N_{on} steps, where the rewards are *not* observable, so the learner collects (s, a, s', d) tuples. The only available online supervision is from a *teacher* who compares behavior across two trajectory segments. A *segment* σ of length H is a contiguous sequence of transitions $\{(s_k, a_k, s_{k+1}, d_k)\}_{k=t}^{t+H-1}$. The learner may submit a pair (σ^1, σ^2) to the teacher, who compares their returns and replies with a label $y \in \{1, 0, \perp\}$, where $y = \mathbb{1}\{\sigma^1 \succ \sigma^2\}$ names the preferred segment and $y = \perp$ is an *abstention* on pairs the teacher cannot order. At most $N_{\max}$ segment pairs may be submitted to the teacher over the whole online phase.

The objective is to output, after N_{on} online steps, a policy π maximizing the expected return $J(\pi)$. Note that the true reward r is used only for evaluation and by the teacher when forming comparisons; it is never observed by the learner. The setting sits between offline-to-online RL, where reward is observable online (Zhou et al., 2025b), and preference-based online RL, where no pretrained actor and critic is available (Lee et al., 2021). Here, actor-critic is available, but the reward is not.

3 Methodology

Our framework warm-starts the agent from the offline checkpoint $(\overline{\pi}, \overline{Q})$ and fine-tunes it online under a limited preference budget.

Online fine-tuning. The online learner is a Soft Actor-Critic agent (Haarnoja et al., 2018) , an off-policy method that trains a critic Q_θ and a policy π_ϕ in the maximum-entropy RL framework. Given a replay buffer $\mathcal{B} = \{(s, a, \widetilde{r}, s')\}$ with a reward signal $\widetilde{r}$, the critic is fit by minimizing the Bellman residual

$$\mathbb{E}_{\tau\sim\mathcal{B}}\Big[\big(Q_\theta(s, a) - \widetilde{r} - \gamma V_*(s')\big)^2\Big], \quad V_*(s') = \mathbb{E}_{a'\sim\pi_\phi}\big[Q_{\theta_*}(s', a') - \eta \log \pi_\phi(a' \mid s')\big], \tag{1}$$

with θ_* the target parameters and η the entropy temperature. The policy is fit by minimizing $\mathbb{E}_{s\sim\mathcal{B},\, a\sim\pi_\phi}[\eta \log \pi_\phi(a \mid s) - Q_\theta(s, a)]$. We use a REDQ critic ensemble (Chen et al., 2021) with an update-to-data ratio greater than one, taking several gradient steps per transition: it allows us to extract enough signal from a small buffer, and the randomized ensemble keeps the resulting high-replay regime from over-estimating values. It also makes a small label budget $N_{\max}$ viable.

Offline pretraining. The actor π_ϕ and critic Q_θ are initialized with pretrained $\overline{\pi}$ and $\overline{Q}$ using offline data $\mathcal{D}$. The checkpoint is produced by Calibrated-Q-Learning (Nakamoto et al., 2023), a conservative

Algorithm 1 ANCOR: ACTIVE PREFERENCE FINE-TUNING WITH CRITIC-GROUNDED REWARD

Require: Checkpoint $(\overline{\pi}, \overline{Q})$, query budget $n_{\max}$, discount factor γ, horizon N_{on}, period N_{int}
Init: Online buffer $\mathcal{B}$, preference buffer $\mathcal{P}$, answerability buffer $\mathcal{H}$, reward model $\widehat{r}_\psi$, count C
for $t = 1$ **to** N_{on} **do**
  Choose $a \sim \pi(\cdot \mid s)$; Observe $s' \sim P(\cdot|s,a)$; Insert (s,a,s') into $\mathcal{B}$
  if $t \bmod N_{\text{int}} = 0$ **and** $C < n_{\max}$ **then**
    $(\mathcal{P}, \mathcal{H}) \leftarrow$ ACTIVESELECT$(\mathcal{B}, \mathcal{P}, \mathcal{H}, \widehat{r}_\psi, b)$; $C \leftarrow C + b$ ▷ Alg. 3
    Retrain $\widehat{r}_\psi$ on $\mathcal{P}$ by minimizing equation 2
    $(\alpha, \beta) \leftarrow$ CRITICGAUGE$(\mathcal{B}, \widehat{r}_\psi)$ ▷ Alg. 2
    $\widetilde{r}(\cdot,\cdot) \leftarrow \alpha\, \widehat{r}_\psi(\cdot,\cdot) + \beta$; Label $\mathcal{B}$ with $\widetilde{r}$
  end if
  if $t > N_w$ **then**
    $(\pi, Q) \leftarrow$ SAC + REDQ update of using $\mathcal{B}$
  end if
end for

offline RL algorithm built on CQL (Kumar et al., 2020). CQL adds a regularizer to the Bellman objective that suppresses Q-values on out-of-distribution actions while compensating for actions present in $\mathcal{D}$. Cal-QL modifies it to additionally require that the learned Q-values upper-bound the state values V_μ of the behavior policy μ, yielding the regularizer

$$\mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi}\big[\max\big(Q_\theta(s,a),\, V_\mu(s)\big)\big] - \mathbb{E}_{s,a\sim\mathcal{D}}\big[Q_\theta(s,a)\big],$$

so that the push-down is masked wherever the Q-function is not *calibrated*. Conservatism alone tends to learn values far below the true return of any valid policy, which causes the policy to unlearn its own initialization when fine-tuning begins; calibration constrains the values to the range of true returns instead (Nakamoto et al., 2023). This ensures that the learned critic $\overline{Q}$ remains approximately Bellman-consistent on $\mathcal{D}$ with respect to the true reward signal r.

Online reward learning. Rewards are not directly observable for transitions collected online, so they must be inferred from pairwise comparisons between trajectory segments. Following Christiano et al. (2017); Lee et al. (2021), we model the teacher's comparisons with a Bradley-Terry-Luce (BTL) likelihood over a learned reward model $\widehat{r}_\psi : \mathcal{T} \times \mathcal{A} \to \mathbb{R}$, i.e.,

$$\mathbb{P}_{\widehat{r}_\psi}\big[\sigma^1 \succ \sigma^2\big] = \mathrm{sigmoid}\big(\widehat{r}_\psi(\sigma^1) - \widehat{r}_\psi(\sigma^2)\big), \text{ where } \widehat{r}_\psi(\sigma) := \sum\nolimits_{(s,a)\in\sigma} \widehat{r}_\psi(s,a).$$

Given a preference buffer $\mathcal{P}$ of triples (σ^1, σ^2, y), the reward model (an ensemble of E MLPs with parameters $\psi = \{\psi_1, \ldots, \psi_E\}$) is fit by minimizing the cross-entropy loss

$$L(\widehat{r}_\psi) = -\sum_{(\sigma^1,\sigma^2,y)\in\mathcal{P}} \overline{y} \log \mathbb{P}_{\widehat{r}_\psi}[\sigma^1 \succ \sigma^2] + (1-\overline{y}) \log \mathbb{P}_{\widehat{r}_\psi}[\sigma^2 \succ \sigma^1], \quad \overline{y} = \begin{cases} y, & y \in \{0,1\}, \\ \frac{1}{2}, & y = \perp, \end{cases} \tag{2}$$

where we treat abstentions as soft labels, since an unordered pair indicates that both segments have similar returns, thereby providing information about the true reward r. Note that the loss is invariant to constant translations $\beta \in \mathbb{R}$, since the log-odds depend only on return differences, while a positive rescaling $\alpha > 0$ sharpens the log-odds. Thus, unconstrained preference training over equal-length segments identifies $\widehat{r}_\psi$ only up to an offset and a scale the preferences cannot pin down.

The pretrained critic $\overline{Q}$ holds values in the units of the true reward r. It is fine-tuned by minimizing the Bellman residual (Eq. (1)), which adds the online reward $\widetilde{r}$ with $\gamma\overline{V}(s')$. Thus, using an arbitrarily placed $\widehat{r}_\psi$ as $\widetilde{r}$ in the online buffer corrupts the Bellman target. We use a *gauged* reward $\widetilde{r}(s,a) = \alpha\, \widehat{r}_\psi(s,a) + \beta$, with properly placed α, β, to ensure $\widetilde{r}$ lies in the same scale as the critic.

3.1 REWARD PLACING VIA CRITIC GAUGE

The pretrained critic $\overline{Q}$ is fit to be approximately Bellman-consistent with respect to the true reward r, so inverting the Bellman equation recovers r point-wise. For a transition (s,a,s'), let

$$\overline{r}(s,a) = \overline{Q}(s,a) - \gamma\, \overline{V}(s'), \text{ where } \overline{V}(s') = \mathbb{E}_{a'\sim\overline{\pi}}\big[\overline{Q}(s',a')\big]. \tag{3}$$

Algorithm 2 CRITICGAUGE

Require: Online buffer $\mathcal{B}$, Checkpoint $(\overline{\pi}, \overline{Q})$, discount factor γ, warm-up N_w

1: $\{(s_i, a_i, s_i')\}_{i=1}^{N_w} \sim \mathcal{B}$; $\overline{r}(s_i, a_i) \leftarrow \overline{Q}(s_i, a_i) - \gamma\, \overline{V}(s_i')$ for all i
2: $\widehat{\mu}$, $\widehat{\sigma} \leftarrow$ mean and std of $\{\widehat{r}_\psi(s_i, a_i)\}$; $\overline{\mu}$, $\overline{\sigma} \leftarrow$ mean and std of $\{\overline{r}(s_i, a_i)\}$
3: $\varrho \leftarrow$ correlation coefficient b/w $\widehat{r}_\psi, \overline{r}$; $w \leftarrow$ shrinkage factor computed using Eq. (4)
4: $\alpha \leftarrow \max\big(0,\ \omega\, \varrho\, \overline{\sigma}/\widehat{\sigma}\big)$; $\beta \leftarrow \overline{\mu} - \alpha\, \widehat{\mu}$
5: **return** α, β

denote the reward the checkpoint $(\overline{\pi}, \overline{Q})$ implies. Computing it costs one evaluation of $\overline{Q}$ and one expectation over $\overline{\pi}$ per transition. Crucially, $\overline{r}$ is in the same scale as r, although it is a noisy target. We fit the scale α and offset β by regressing the implied reward $\overline{r}$ on the learned reward $\widehat{r}_\psi$ over a warm-up batch $\{(s_i, a_i, s_i')\}_{i=1}^{N_w}$, randomly sampled from $\mathcal{B}$. Let $(\overline{\mu}, \overline{\sigma})$ be the sample mean and standard deviation of $\{\overline{r}(s_i, a_i)\}$, and $(\widehat{\mu}, \widehat{\sigma})$ be those of $\{\widehat{r}_\psi(s_i, a_i)\}$. The ordinary least squares slope is $\varrho\, \overline{\sigma}/\widehat{\sigma}$, where ϱ denotes the correlation coefficient between $\widehat{r}_\psi$ and $\overline{r}$. Since N_w is typically small and both $\widehat{r}_\psi, \overline{r}$ are noisy, the slope could have high variance. Thus, we shrink it using the disagreement over the reward ensemble $\{\psi_1, \ldots, \psi_E\}$. To this end, let ϱ_j denote the correlation coefficient considering when $\psi = \psi_j$ and $\overline{\varrho}$ be their mean. The shrinkage factor

$$\omega = \max\Big(0, 1 - \tfrac{\tau^2 + 1/N_w}{\varrho^2}\Big), \text{ where } \tau^2 = \tfrac{1}{E(E-1)} \sum\nolimits_{j=1}^{E} (\varrho_j - \overline{\varrho})^2\,, \tag{4}$$

quantifies how much of ϱ is signal. Its numerator collects two independent sources of error: τ^2 is the spread of $\overline{\varrho}$ across the ensemble, large when the reward models disagree about how they relate to $\overline{r}$, and $1/N_w$ is the finite-sample uncertainty of the correlation, taken over an effective sample size that discounts the strong autocorrelation of the consecutive warm-up transitions. Dividing their sum by ϱ^2 weighs the noise against the observed correlation: ω is near one when ϱ is large relative to its uncertainty, and falls to zero when the correlation is indistinguishable from the noise. With this, we define the scale and offset of the gauge as

$$\alpha = \max(0,\ \omega\, \varrho\, \overline{\sigma}/\widehat{\sigma}), \qquad \beta = \overline{\mu} - \alpha\, \widehat{\mu}. \tag{5}$$

As preferences accumulate and the ensemble converges, ω rises, and the scale is applied with no threshold. The offset matches means, so $\widetilde{r}$ sits at the same average level as $\overline{r}$ on warm-up data.

The online phase lasts N_{on} steps, of which the first N_w are warm-up, during which the agent acts under $\overline{\pi}$ and no queries are issued. Thereafter, a feedback session is triggered every N_{int} steps, and whenever the reward model is updated, we relabel the entire buffer with $\widetilde{r}$. A counter tracks the number of comparisons labeled by the teacher so far, and querying stops once it reaches the budget $N_{\max}$. Hence, it is crucial to decide which pairs of segments to compare.

3.2 ACTIVE LABELING VIA INFORMATION MAXIMIZATION

We first reformulate the problem of selecting which segment pairs to label from a given pool as a Bayesian experimental design problem with a Gaussian posterior over the reward model's final-layer weights. Then, we design a greedy algorithm that solves it with an approximation guarantee.

Let $\varphi_\psi(\sigma) \in \mathbb{R}^d$ denote the (frozen) feature representation of the penultimate layer induced by the reward model $\widehat{r}_\psi$ for a trajectory segment σ, so that σ is scored as $\widehat{r}_\psi(\sigma) = w^\top \varphi_\psi(\sigma)$, where $w \in \mathbb{R}^d$ collects the weights of the final linear layer of $\widehat{r}_\psi$. Under the Bradley-Terry (BTL) model, the teacher's preference between a pair (σ^1, σ^2) is a Bernoulli random variable, whose parameter depends on w only through the feature differences, henceforth called the *contrast* vector,

$$z = \varphi_\psi(\sigma^1) - \varphi_\psi(\sigma^2), \quad \text{so that} \quad \mathbb{P}_{\widehat{r}_\psi}\big[\sigma^1 \succ \sigma^2 \mid w\big] = \text{sigmoid}\big(w^\top z\big)\,.$$

Thus, two different segment pairs with the same contrast have the same probability of preference.

We maintain a Gaussian belief $w \sim \mathcal{N}(\mu_w, \Sigma_w)$ over the final layer weights. The mean vector μ_w is initialized as $\widehat{r}_\psi$'s final weights, so that the first round of selection is performed using the information the reward model has already accumulated. The covariance matrix is initialized isotropically, $\Sigma_w = \lambda^{-1} I$ for some $\lambda > 0$. The Gaussian prior is not conjugate to the BTL (logistic) likelihood, so the exact posterior update is intractable. We therefore use an assumed-density update (Laplace approximation) per labeled pair, which is the standard Gaussian projection of the true posterior.

Lemma 3.1 (One-step Laplace update). *Let $w \sim \mathcal{N}(\mu_w, \Sigma_w)$ and $y \sim \mathrm{Bernoulli}(\mathrm{sigmoid}(w^\top z))$ be the label of a pair with contrast $z \in \mathbb{R}^d$. Then, conditioned on y, the Gaussian approximation of the posterior of w has covariance matrix and mean vector*

$$\Sigma'_w = \left(\Sigma_w^{-1} + c\, zz^\top\right)^{-1} = \Sigma_w - \frac{c}{1 + c\, z^\top \Sigma_w z} (\Sigma_w z)(\Sigma_w z)^\top, \qquad \mu'_w = \mu_w + \Sigma'_w\, (y - p)\, z,$$

where $p = \mathrm{sigmoid}(\mu_w^\top z)$ and $c = p(1-p)$ denote, respectively, the mean and variance of y under prior belief $w = \mu_w$.

Some observations are in order. The Hessian of the negative log-likelihood of y, labeled using the prior model μ_w, evaluates to $\nabla^2 \ell(\mu_w) = c\, zz^\top$ – the Fisher information matrix. It measures how much information we get by labeling a pair with contrast z. It is maximized at $p = \frac{1}{2}$ when the prior is most uncertain and vanishes as $p \to 0$ or $p \to 1$: a pair whose outcome the prior model predicts confidently carries no information, regardless of the contrast z. Moreover, the label y enters only in the posterior mean. When the teacher returns a tie ($y = \perp$), we set it to $\frac{1}{2}$, the expected label for an indifferent comparison under the BTL model. The posterior covariance is a function of z and the current belief alone. The reduction in posterior uncertainty by labeling a pair is therefore *computable before the label is observed*, which makes it usable as a score for selecting informative pairs.

3.2.1 Selecting Informative Pairs

Define the *variance-scaled contrast* $\zeta_i = \sqrt{c_i} z_i$ for the i-th pair so that the covariance update reads $\Sigma'_w = \Sigma_w - \frac{(\Sigma_w \zeta_i)(\Sigma_w \zeta_i)^\top}{1 + \zeta_i^\top \Sigma_w \zeta_i}$. Write $\mathcal{C} = \{1, \ldots, n\}$ for the index set of the pool, and for $\mathcal{T} \subset \mathcal{C}$, write $\Sigma_\mathcal{T} = \left(\Sigma_w^{-1} + \sum_{i \in \mathcal{T}} \zeta_i \zeta_i^\top\right)^{-1}$ for the posterior covariance matrix that would result from labeling all pairs indexed by $\mathcal{T}$, with $\Sigma_\emptyset = \Sigma_w$. For a segment σ_j, the score $w^\top \varphi_\psi(\sigma_j)$ is a linear functional of w, so its posterior variance after labeling $\mathcal{T}$ is $\varphi_\psi(\sigma_j)^\top \Sigma_\mathcal{T} \varphi_\psi(\sigma_j)$. Since $\Sigma_{\mathcal{T}'} \preceq \Sigma_\mathcal{T}$ in the Lowener order whenever $\mathcal{T}' \supseteq \mathcal{T}$, posterior uncertainty is monotonically non-increasing in the labeled set.

The BTL likelihood sees only differences of features, so adding a common vector to every $\varphi_\psi(\sigma)$ leaves every contrast and hence the likelihood unchanged. The component of $\Sigma_\mathcal{T}$ along this direction is never reduced by any comparison and would contribute an irreducible constant to every candidate's score. We therefore center the features in the pool, $\widetilde{\varphi}_j = \varphi_\psi(\sigma_j) - \overline{\varphi}$, where $\overline{\varphi}$ is the mean feature over $2n$ pool segments, to remove the non-identifiable direction and leaves an objective that measures uncertainty only in the relative scores that preference comparisons can inform. With this, we define a set function $f : 2^\mathcal{C} \to \mathbb{R}$ as

$$f(\mathcal{T}) = -\tfrac{1}{2n} \sum_{j=1}^{2n} \widetilde{\varphi}_j^\top \Sigma_\mathcal{T} \widetilde{\varphi}_j = -\operatorname{tr}\big(\Sigma_\mathcal{T} M\big), \qquad M = \tfrac{1}{2n} \sum_{j=1}^{2n} \widetilde{\varphi}_j \widetilde{\varphi}_j^\top \tag{6}$$

where $M \succeq 0$ is the feature covariance of the pool. We solve the design problem of maximizing f, or equivalently, minimizing average posterior uncertainty, over sets of size at most b.

Maximizing f over all subsets of size b is combinatorial, with $\binom{n}{b}$ candidates. We solve it greedily: starting from $\mathcal{T} = \emptyset$, $\mu_\mathcal{T} = \mu_w$ and $\Sigma_\mathcal{T} = \Sigma_w$, at each step $t \in \{1, \ldots, b\}$, we add the pair with the largest marginal gain over already labeled pairs.

Lemma 3.2 (Marginal gain of f and posterior update). *For any set $\mathcal{T}$, let μ_τ and Σ_τ be the posterior mean and covariance of w after labeling all segment pairs indexed by $\mathcal{T}$. Then, for any pair $i \notin \mathcal{T}$,*

$$u_i(\mathcal{T}) := f(\mathcal{T} \cup \{i\}) - f(\mathcal{T}) = \frac{\zeta_i^\top \Sigma_\mathcal{T} M \Sigma_\mathcal{T} \zeta_i}{1 + \zeta_i^\top \Sigma_T \zeta_i} = \frac{c_i \cdot \frac{1}{2n} \sum_{j=1}^{2n} \big((\Sigma_\mathcal{T} z_i)^\top \widetilde{\varphi}_j\big)^2}{1 + c_i\, z_i^\top \Sigma_\mathcal{T} z_i}, \tag{7}$$

where $c_i = p_i(1-p_i)$ and $p_i = \mathrm{sigmoid}(\mu_\mathcal{T}^\top z_i)$. Moreover, let $y_i = \{0, 1, \perp\}$ be the label of i-th pair. Conditioned on this, the posterior covariance and mean are updated as

$$\Sigma_\mathcal{T} \leftarrow \Sigma_\mathcal{T} - \frac{c_i}{1 + c_i z_i^\top \Sigma_\mathcal{T} z_i} (\Sigma_\mathcal{T} z_i)(\Sigma_\mathcal{T} z_i)^\top, \quad \mu_\mathcal{T} \leftarrow \mu_\mathcal{T} + \Sigma_\mathcal{T}\, (\overline{y}_i - p_i)\, z_i\,, \tag{8}$$

where $\overline{y}_i = 1/2$ if $y_i = \perp$, and $\overline{y}_i = y_i$ otherwise.

The numerator of the score $u_i(\mathcal{T})$ is large when the direction probed by pair i is one along which *many* pool segments' scores are uncertain, so a label of the pair is broadly useful, while the weight c_i suppresses pairs that the model already predicts confidently. The denominator encodes how much uncertainty that a single binary label can resolve along z_i, so the value of querying does not grow without bound as z_i becomes more uncertain.

Algorithm 3 ACTIVESELECT

Require: Online buffer $\mathcal{B}$, preference buffer $\mathcal{P}$, answerability buffer $\mathcal{H}$, reward model $\widehat{r}_\psi$, batch size b, Checkpoint $(\overline{\pi}, \overline{Q})$, discount factor γ, pool size n

Train $p_{\text{ans}}(\Delta) = \text{sigmoid}(a + b\,\Delta)$ on $\mathcal{H}$;
$\varphi_\psi(\sigma) \leftarrow$ penultimate layer features of $\widehat{r}_\psi(\sigma)$; $\mu_w \leftarrow$ final layer weights of $\widehat{r}_\psi$; $\Sigma_w \leftarrow \lambda^{-1} I$
for $(\sigma^1, \sigma^2, \overline{y}) \in \mathcal{P}$ **do**
  $z \leftarrow \varphi_\psi(\sigma^1) - \varphi_\psi(\sigma^2)$; $p \leftarrow \text{sigmoid}(\mu_w^\top z)$; $c \leftarrow p(1-p)$
  $\Sigma_w \leftarrow \Sigma_w - \dfrac{c}{1 + c\, z^\top \Sigma_w z} (\Sigma_w z)(\Sigma_w z)^\top$; $\mu_w \leftarrow \mu_w + \Sigma_w(\overline{y} - p)\, z$
end for
$(\sigma_i^1, \sigma_i^2)_{i=1}^n \sim \mathcal{B}$; $\mathcal{C} \leftarrow \{1, \ldots, n\}$; $\mathcal{T} \leftarrow \emptyset$; $\mu_{\mathcal{T}} \leftarrow \mu_w$; $\Sigma_{\mathcal{T}} \leftarrow \Sigma_w$
for $t = 1, \ldots, b$ **do**
  Compute $u_j(\mathcal{T})$ from Eq. (7) and Δ_j from Eq. (9) for each $j \in \mathcal{C} \setminus \mathcal{T}$
  $i \leftarrow \arg\max_{j \in \mathcal{C} \setminus \mathcal{T}} p_{\text{ans}}(\Delta_j) u_j$; $\mathcal{T} \leftarrow \mathcal{T} \cup \{i\}$
  $y_i \leftarrow \mathbb{1}\{\sigma_i^1 \succ \sigma_i^2\}$ or $\perp$; Insert $(\Delta_i, \mathbb{1}[y_i \neq \perp])$ into $\mathcal{H}$
  $\overline{y}_i \leftarrow \frac{1}{2}$ if $y_i = \perp$, else $\overline{y}_i \leftarrow y_i$; Insert $(\sigma_i^1, \sigma_i^2, \overline{y}_i)$ into $\mathcal{P}$
  Update $\Sigma_{\mathcal{T}}$ and $\mu_{\mathcal{T}}$ from Eq. (8)
end for
return $\mathcal{P}, \mathcal{H}$

3.2.2 ANSWERABILITY WEIGHTING

The score $u_i(\mathcal{T})$ measures the incremental value of labeling a pair i over an already labeled set $\mathcal{T}$ *conditional on receiving a label*. The teacher, however, may decline to answer ($y_i = \perp$), and is more likely to do so for segment pairs with a small return difference. We therefore weight the scores with a calibrated probability of answering. First, we model answering as an independent Bernoulli event with probability $p_{\text{ans}}(\Delta)$, where

$$\Delta = \left|\overline{r}(\sigma^1) - \overline{r}(\sigma^2)\right|, \quad \text{with } \overline{r}(\sigma) = \sum\nolimits_{(s,a,s') \in \sigma} \left(\overline{Q}(s,a) - \gamma\, \overline{V}(s')\right), \tag{9}$$

denotes the return difference of a pair, implied by the checkpoint $(\overline{\pi}, \overline{Q})$. Then, for a set $\mathcal{T}$, we define the *answerability-weighted* objective $f_{\text{ans}}(\mathcal{T}) = \mathbb{E}\big[f(\mathcal{T}_0)\big]$, where $\mathcal{T}_0$ is the subset of pairs the teacher actually answers, each $i \in \mathcal{T}$ answered independently with probability $p_{\text{ans}}(\Delta_i)$.

Lemma 3.3 (Marginal gain of f_{ans})**.** *For any set $\mathcal{T}$ and any $i \notin \mathcal{T}$,*

$$f_{\text{ans}}(\mathcal{T} \cup \{i\}) - f_{\text{ans}}(\mathcal{T}) = p_{\text{ans}}(\Delta_i) \cdot \mathbb{E}\big[u_i(\mathcal{T}_0)\big]\,.$$

The gain involves an expectation over $2^{|\mathcal{T}|}$ realizations of queries answered earlier. So, we use a surrogate $p_{\text{ans}}(\Delta_i)\, u_i(\mathcal{T})$ as our selection score, which coincides with the gain when all previously selected pairs were answered ($\mathcal{T}_0 = \mathcal{T}$). At each step t, we select the pair

$$i^* = \arg\max_{i \in \mathcal{C} \setminus \mathcal{T}} p_{\text{ans}}(\Delta_i) \cdot u_i(\mathcal{T}), \;\; \text{where } \mathcal{C} = \{1, \ldots, n\}. \tag{10}$$

We estimate p_{ans} as a one-dimensional logistic in the return gap, $p_{\text{ans}}(\Delta) = \text{sigmoid}(a + b\Delta)$, refit each round from both answered and tied outcomes, which are accumulated in the answerability buffer $\mathcal{H} = \{(\Delta, \mathbb{1}[y \neq \perp])\}$. Until outcomes of both kinds exist, we set $p_{\text{ans}}(\Delta) = \text{sigmoid}(\Delta)$. The objective f_{ans} is monotone and approximately submodular, with a submodularity ratio bounded below by $\gamma = \frac{1}{1 + \max_i \|z_i\|^2/(4\lambda)}$. The bound is governed by the most informative single contrast relative to the prior scale λ. Writing $g_{\text{ans}}(\mathcal{T}) = f_{\text{ans}}(\mathcal{T}) - f_{\text{ans}}(\emptyset)$ for the pool variance reduction it achieves, which shares the marginal gains and hence the selection rule of f_{ans}, greedy selection attains a $(1 - e^{-\gamma})$-approximation to the optimal batch, independent of the batch size b (Appendix B.4).

4 EXPERIMENTS

We compare ANCOR with WSRL (Zhou et al., 2025b), which starts from the same offline checkpoint and observes online rewards, and PEBBLE (Lee et al., 2021), which learns from preferences without an offline checkpoint. These comparisons assess performance relative to reward-supervised fine-tuning and preference learning without an offline prior. We evaluate on eight tasks: kitchen-mixed

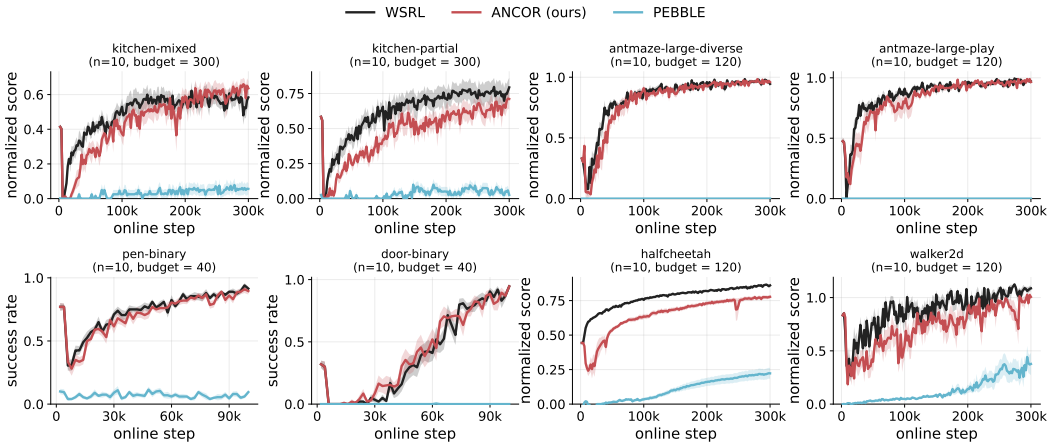


Figure 1: Comparison of ANCOR (red) with WSRL (black) and PEBBLE (blue) on eight tasks. ANCOR uses preference feedback, WSRL observes online rewards, and PEBBLE learns from preferences without an offline checkpoint. Panel titles give feedback budgets. Curves show mean ± one standard error over $n = 10$ seeds.

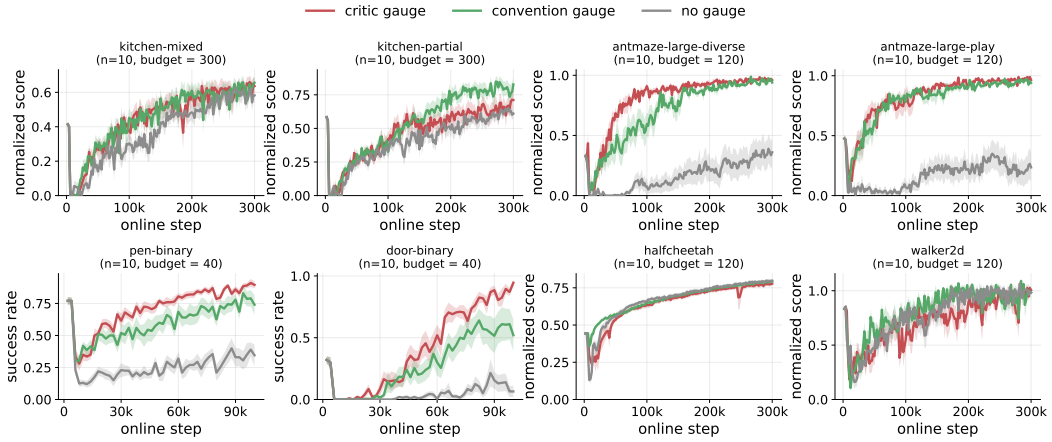


Figure 2: Reward calibration on eight tasks: critic gauge (red), convention gauge (green), and no gauge (gray). The convention gauge uses rewarded offline data; the critic gauge uses the frozen checkpoint and online transitions. Budgets are shown above each panel. Mean ± one standard error over 10 seeds.

and kitchen-partial (Kitchen), antmaze-large-diverse and antmaze-large-play (AntMaze), pen-binary and door-binary (Adroit), and halfcheetah and walker2d (locomotion). We initialize from a Cal-QL (Nakamoto et al., 2023) offline checkpoint and use SAC with a REDQ (Chen et al., 2021) critic ensemble. We perform online fine-tuning for 100,000 environment steps in Adroit and 300,000 steps in all other tasks. A simulated teacher supplies preferences every 5,000 steps (see Appendix C.3 for details). Query budgets are 300 for Kitchen, 40 for Adroit, and 120 for all other tasks. We report D4RL-normalized scores (Fu et al., 2021), except in Adroit, where we report success rates.

Results. Figure 1 shows that the proposed algorithm ANCOR largely closes the performance gap with online reward-supervised WSRL (the skyline) across all eight tasks, even matching WSRL on some tasks (kitchen-mixed, AntMaze, Adroit). Compared with PEBBLE (the baseline), which receives the same preference budget, ANCOR performs substantially better. These results show that fine-tuning an offline prior with limited preferences can approach reward-supervised performance while substantially outperforming preference learning from scratch.

4.1 Ablation Study

Reward Gauge. Figure 2 compares ANCOR's performance while running with our proposed critic gauge as opposed to running with a privileged conventional gauge and with uncalibrated online rewards. The conventional gauge has access to ground-truth offline rewards and thus serves as a reference (see Appendix C.1 for details). We observe that the critic gauge is competitive with this privileged reference across most tasks and performs better on Adroit, despite lacking access to offline

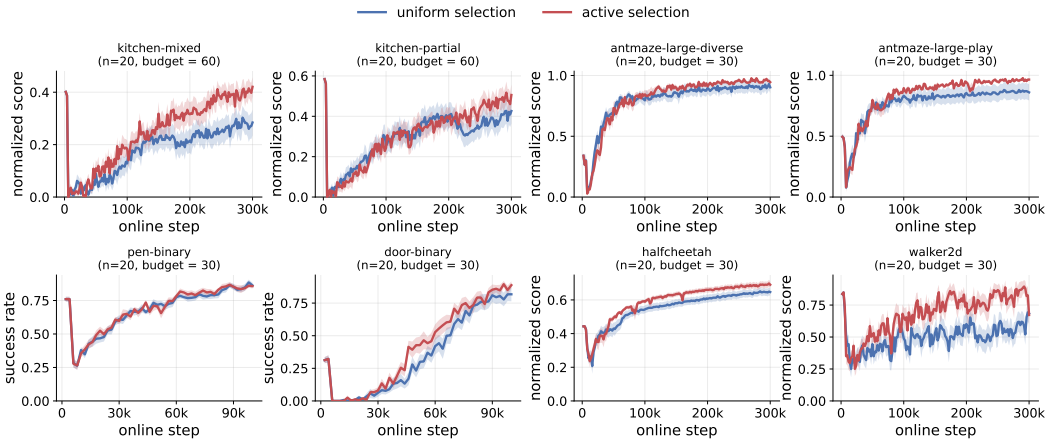


Figure 3: Active (red) versus uniform (blue) query selection with the critic gauge. Budgets are 60 queried pairs on Kitchen and 30 elsewhere. Mean ± one standard error over 20 seeds.

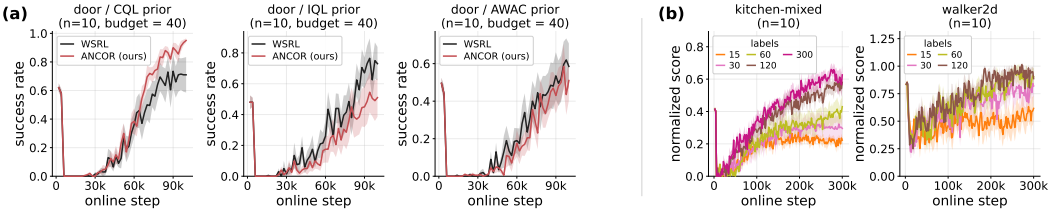


Figure 4: (a) Effect of the offline prior (CQL, IQL, AWAC) on door-binary. (b) Effect of the preference-query budget on kitchen-mixed (left) and walker2d (right). All plots are mean ± one standard error over 10 seeds.

reward. Calibration yields large gains on AntMaze and Adroit, modest gains on Kitchen, and little change on locomotion compared to no calibration.

Query Selection method. Figure 3 compares ANCOR's performance while running with our proposed active query selection strategy as opposed to selecting queries uniformly. We use a lower query budget (60 queries on Kitchen and 30 elsewhere) to isolate the effect of the sampling strategy. We observe that Active selection improves the mean learning curves most clearly on Kitchen and Walker2d, with smaller gains on Door, HalfCheetah, and antmaze-large-play, and behaves similarly on Pen and antmaze-large-diverse. We conclude that the benefit of active selection is task dependent.

Offline prior. Figure 4(a) compares ANCOR and WSRL on door-binary under different offline checkpoints, CQL (Kumar et al., 2020), IQL (Kostrikov et al., 2022) and AWAC (Nair et al., 2020). ANCOR performs similarly to WSRL across all these checkpoints, indicating that its relative performance against WSRL is checkpoint-agnostic.

Preference Budget. Figure 4(b) compares ANCOR's performance under different preference budgets on kitchen-mixed and walker2d. We observe that larger budgets improve average performance on both tasks. Kitchen continues to benefit from up to 300 queries, while Walker2d shows smaller gains when the budget is increased from 60 to 120 queries.

5 Conclusion

We studied online fine-tuning in which the checkpoint is the only artifact of pretraining, and the only supervision is a simulated labeler that compares short segments and may abstain. Two obstacles arise. A reward learned from preferences is identified only up to scale and offset, and writing it as is into the replay buffer overwrites the values the pretrained critic holds; our critic gauge recovers the placement from the checkpoint itself, without the offline data or reward. Labels are scarce, and the most uncertain pairs are often unanswerable; our active selection rule weights expected posterior-variance reduction by a calibrated estimate of answerability. On eight tasks, ANCOR closely matches reward-supervised WSRL and outperforms PEBBLE by a wide margin.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing methods, namely writing and debugging parts of the experimental code. We have not used generative AI tools to develop the theoretical models or conceptual framework, formulate mathematical claims, prove or assist in proving mathematical claims, propose or refine hypotheses, design the research methodology or experiments, or interpret results, and the remaining required-disclosure tasks (generating synthetic datasets, translation, cleaning and reformatting datasets, and qualitative or thematic data analysis) are not applicable to this work. Additionally, we used generative AI tools to edit the paper for readability and to draft LaTeX formatting, such as tables, cross-references, and figure code. We have reviewed all AI-assisted work. All AI-assisted code was verified and tested for correctness by the authors, and all AI-edited text was reviewed by the authors for accuracy and fidelity to the intended claims. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The full procedure is given in Algorithms 1–3. Appendix C.4 lists all hyperparameters, checkpoint settings, query budgets, teacher definitions, and baseline configurations used in every experiment, and Appendix C.1 gives the exact gauge estimator. All tasks use publicly available D4RL datasets. Source code to reproduce results is submitted as supplementary material.

REFERENCES

Akhil Agnihotri, Rahul Jain, Deepak Ramachandran, and Zheng Wen. Online bandit learning with offline preference data. *CoRR*, abs/2406.09574, 2024. doi: 10.48550/ARXIV.2406.09574. URL https://doi.org/10.48550/arXiv.2406.09574.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1577–1594. PMLR, 2023. URL https://proceedings.mlr.press/v202/ball23a.html.

Erdem Biyik, Malayandi Palan, Nicholas C. Landolfi, Dylan P. Losey, and Dorsa Sadigh. Asking easy questions: A user-friendly approach to active reward learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1177–1190. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/b-iy-ik20a.html.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chenyang Cao, Miguel Rogel Garcia, Mohamed Nabail, Xueqian Wang, and Nicholas Rhinehart. Residual reward models for preference-based reinforcement learning. *CoRR*, abs/2507.00611, 2025. doi: 10.48550/ARXIV.2507.00611. URL https://doi.org/10.48550/arXiv.2507.00611.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. Randomized ensembled double q-learning: Learning fast without a model. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=AY8zfZm0tDd.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In Lise Getoor and Tobias Scheffer (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 1057–1064. Omnipress, 2011. URL https://icml.cc/2011/papers/542_icmlpaper.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023. URL https://arxiv.org/abs/2303.03378.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021. URL https://arxiv.org/abs/2004.07219.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

Donald Joseph Hejna and Dorsa Sadigh. Few-shot preference learning for human-in-the-loop rl. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2014–2025. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/iii23a.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 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 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Kimin Lee, Laura M Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6152–6163. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/lee21i.html.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1702–1712. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/lee22d.html.

Chenran Li, Chen Tang, Haruki Nishimura, Jean Mercat, Masayoshi TOMIZUKA, and Wei Zhan. Residual q-learning: Offline and online policy customization without value. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 61857–61869. Curran Associates, Inc., 2023a. doi: 10.52202/075280-2704. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c2e4cebba2fdb3dac7d2022421062765-Paper-Conference.pdf.

Jianxiong Li, Xiao Hu, Haoran Xu, Jingjing Liu, Xianyuan Zhan, and Ya-Qin Zhang. PROTO: iterative policy regularized offline-to-online reinforcement learning. *CoRR*, abs/2305.15669, 2023b. doi: 10.48550/ARXIV.2305.15669. URL https://doi.org/10.48550/arXiv.2305.15669.

Lu Li, Tianwei Ni, Yihao Sun, and Pierre-Luc Bacon. The three regimes of offline-to-online reinforcement learning. *CoRR*, abs/2510.01460, 2025a. doi: 10.48550/ARXIV.2510.01460. URL https://doi.org/10.48550/arXiv.2510.01460.

Qiyang Li, Zhiyuan (Paul) Zhou, and Sergey Levine. Reinforcement learning with action chunking. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 55518–55553. Curran Associates, Inc., 2025b. doi: 10.52202/085713-1856. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/50348e8f9aef984abe0ea1ec2a326f78-Paper-Conference.pdf.

Mael Macuglia, Paul Friedrich, and Giorgia Ramponi. Fine-tuning behavioral cloning policies with preference-based reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=oIiQZfnSxP.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *CoRR*, abs/2006.09359, 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 A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 62244–62269. Curran Associates, Inc., 2023. doi: 10.52202/075280-2719. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c44a04289beaf0a7d968a94066a1d696-Paper-Conference.pdf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017. doi: 10.15607/RSS.2017.XIII.053.

Yongjae Shin, Jeonghye Kim, Whiyoung Jung, Sunghoon Hong, Deunsol Yoon, Youngsoo Jang, Geon-Hyeong Kim, Jongseong Chae, Youngchul Sung, Kanghoon Lee, and Woohyung Lim. Online pre-training for offline-to-online reinforcement learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/shin25c.html.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Shenzhi Wang, Qisen Yang, Jiawei Gao, Matthieu Lin, HAO CHEN, Liwei Wu, Ning Jia, Shiji Song, and Gao Huang. Train once, get a family: State-adaptive balances for offline-to-online reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47081–47104. Curran Associates, Inc., 2023. doi: 10.52202/075280-2039. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9318763d049edf9a1f2779b2a59911d3-Paper-Conference.pdf.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952, 2023. doi: 10.1109/ICCV51070.2023.01100.

Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. A perspective of q-value estimation on offline-to-online reinforcement learning. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 16908–16916. AAAI Press, 2024.

doi: 10.1609/AAAI.V38I15.29633. URL https://doi.org/10.1609/aaai.v38i15.29633.

Zhiyuan Zhou, Pranav Atreya, Abraham Lee, Homer Rich Walke, Oier Mees, and Sergey Levine. Autonomous improvement of instruction following skills via foundation models. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4805–4825. PMLR, 06–09 Nov 2025a. URL https://proceedings.mlr.press/v270/zhou25b.html.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=HN0CYZbAPw.

Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11): 13344–13362, 2023. doi: 10.1109/TPAMI.2023.3292075.

A Extended Related Work

Offline-to-Online Reinforcement Learning Offline-to-online (O2O) reinforcement learning trains a policy on a fixed dataset and then continues to improve it through interaction with the environment. Naively fine-tuning an offline-pretrained actor-critic with off-policy RL often causes immediate performance degradation, often termed *policy collapse*, followed by slow or stalled improvement (Nair et al., 2020; Lee et al., 2022; Nakamoto et al., 2023; Zhang et al., 2024).

Zhang et al. (2024) diagnose the root cause as inaccurate Q-value estimation inherited from offline training, proposing perturbed value updates and increased critic update frequency. Nakamoto et al. (2023) argue that standard conservative offline objectives (e.g., CQL) produce miscalibrated Q-functions that impede fine-tuning, and propose Cal-QL to produce conservative yet well-calibrated initializations. Zhou et al. (2025b) attribute the widely adopted practice of retaining offline data during online fine-tuning primarily to preventing early value-function divergence, and show that a short warmup phase with frozen-policy rollouts can entirely substitute for data retention. Shin et al. (2025) observe that directly fine-tuning offline-pretrained value functions can underperform random initialization, motivating an intermediate online pre-training phase to prepare the critic for adaptation.

Orthogonal to critic-focused solutions, several methods address O2O through replay and constraint design. Balanced replay with pessimistic Q-ensembles (Lee et al., 2022) prioritizes near-on-policy offline samples to reduce bootstrap error from distribution shift. RLPD (Ball et al., 2023) demonstrates that minimal modifications to standard off-policy methods suffice when offline data is mixed into online training. FamO2O (Wang et al., 2023) trains a family of policies under varying constraint intensities and selects among them in a state-adaptive manner. PROTO (Li et al., 2023b) applies an iteratively relaxed trust-region regularizer to bridge any offline policy with online fine-tuning. More recently, action chunking has been applied to O2O fine-tuning to enable temporally coherent exploration in long-horizon sparse-reward tasks (Li et al., 2025b).

(Li et al., 2025a): offline conservatism prevents early collapse but can cap online improvement, and different O2O regimes require preserving whichever prior knowledge source, whether offline data or the pretrained policy, is stronger. All of these methods assume the same scalar reward is available offline and online. We do not observe the online reward, and the agent must adapt both its policy and its reward model during deployment.

Preference-Based Reinforcement Learning Preference-based RL (PbRL) replaces scalar reward signals with pairwise human preferences over trajectory segments, learning a reward model from comparisons and optimizing a policy against it (Christiano et al., 2017; Lee et al., 2021). Most PbRL methods operate either in a purely online setting, actively querying preferences during interaction with the environment, or in a purely offline setting, where preferences are collected from a fixed dataset.

Few works combine preference-based reward learning with an offline-to-online transition in multi-step control. Agnihotri et al. (2024) study warm-starting online learning from offline preference data, but only in the bandit (stateless) setting. Macuglia et al. (2026) propose a two-stage pipeline with offline pretraining followed by online preference-based fine-tuning, but rely on behavioural cloning rather than RL in the offline phase.

Reward Mismatch and Reward Transfer A small body of work considers settings where a proxy or prior reward differs from the true task reward. Residual reward models (Cao et al., 2025) decompose the true reward into a known prior and a learned residual, addressing the mismatch between a proxy reward and the true objective in preference-based RL. Residual Q-learning (Li et al., 2023a) customises a prior policy for an additional task reward without requiring the prior policy's reward or value function. Neither addresses the case where the reward becomes unobservable at the offline-to-online boundary.

In the broader transfer RL literature, reward adaptation has been studied under domain shift (Zhu et al., 2023), but these works typically assume access to some form of online reward signal (even if sparse or differently shaped) rather than addressing the complete absence of reward. We fine-tune an offline actor-critic checkpoint with preferences as the only supervision, and place the learned reward in the critic's units.

B MISSING PROOFS

B.1 PROOF OF LEMMA 3.1

We restate the claim. Let $w \sim \mathcal{N}(\mu_w, \Sigma_w)$ with $\Sigma_w \succ 0$, and let a labelled pair with contrast $z \in \mathbb{R}^d$ and soft label $y \in [0,1]$ be observed under the Bradley–Terry likelihood. Write $p = \mathrm{sigmoid}(z^\top \mu_w)$ and $c = p(1-p)$. Then, the Laplace approximation to the posterior is $\mathcal{N}(\mu'_w, \Sigma'_w)$ with

$$\Sigma'_w = \Sigma_w - \frac{c}{1 + c\, z^\top \Sigma_w z} (\Sigma_w z)(\Sigma_w z)^\top, \qquad \mu'_w = \mu_w + \Sigma'_w\, (y - p)\, z.$$

Proof. For a label $y \in [0,1]$ the negative log-likelihood of the Bradley–Terry observation with contrast z is

$$\ell(w) = -y \log\big(\mathrm{sigmoid}(w^\top z)\big) - (1-y) \log\big(1 - \mathrm{sigmoid}(w^\top z)\big).$$

Write $s(u) = \mathrm{sigmoid}(u)$ and use $s'(u) = s(u)(1 - s(u))$. Applying the chain rule with $u = w^\top z$, so $\nabla_w u = z$,

$$\nabla \ell(w) = \Big[-y\, \frac{s'(u)}{s(u)} + (1-y)\, \frac{s'(u)}{1 - s(u)}\Big] z = \big[-y(1 - s(u)) + (1-y) s(u)\big] z = \big(s(w^\top z) - y\big) z,$$

where the middle equality substitutes $s'(u) = s(u)(1 - s(u))$. Differentiating once more,

$$\nabla^2 \ell(w) = s'(w^\top z)\, z z^\top = s(w^\top z)\big(1 - s(w^\top z)\big)\, z z^\top.$$

Note that $\nabla^2 \ell(w) \succeq 0$ for every w, so ℓ is convex; and that the Hessian does not involve y. Evaluating at $w = \mu_w$ gives $\nabla \ell(\mu_w) = (p - y) z$ and $\nabla^2 \ell(\mu_w) = c\, z z^\top$, the Fisher information of a single Bernoulli observation at the current mean.

The negative log-posterior, up to an additive constant, is

$$\Psi(w) = \ell(w) + \tfrac{1}{2} (w - \mu_w)^\top \Sigma_w^{-1} (w - \mu_w).$$

This is not quadratic in w, so the posterior is not Gaussian.

Replace $\ell(w)$ by its second-order Taylor expansion about the current mean μ_w, to get

$$\ell(w) \approx \ell(\mu_w) + (p - y)\, z^\top (w - \mu_w) + \tfrac{1}{2} (w - \mu_w)^\top \big(c\, z z^\top\big)(w - \mu_w),$$

Writing $\delta = w - \mu_w$ and dropping constants, the approximate negative log-posterior is

$$\widetilde{\Psi}(\delta) = (p - y)\, z^\top \delta + \tfrac{1}{2}\, \delta^\top \big(\Sigma_w^{-1} + c\, z z^\top\big) \delta.$$

This is quadratic in δ with a positive definite Hessian (a sum of a positive definite and a positive semidefinite matrix), so $\exp(-\widetilde{\Psi})$ is an unnormalized Gaussian density. Completing the square,

$$\widetilde{\Psi}(\delta) = \tfrac{1}{2} \big(\delta - a\big)^\top A \big(\delta - a\big) + \text{const}, \quad A = \Sigma_w^{-1} + c\, z z^\top, \quad a = -A^{-1}(p - y) z.$$

Hence the approximating Gaussian has covariance $\Sigma'_w = A^{-1}$ and mean $\mu'_w = \mu_w + a = \mu_w + \Sigma'_w (y - p) z$. It remains to put both in the stated form.

The Sherman-Morrison identity states that for invertible $B \in \mathbb{R}^{d \times d}$ and vectors u, v with $1 + v^\top B^{-1} u \neq 0$,

$$\big(B + u v^\top\big)^{-1} = B^{-1} - \frac{B^{-1} u\, v^\top B^{-1}}{1 + v^\top B^{-1} u}.$$

Apply it with $B = \Sigma_w^{-1}$, $u = c\, z$ and $v = z$, so that $B + u v^\top = \Sigma_w^{-1} + c\, z z^\top = A$ and $B^{-1} = \Sigma_w$:

$$\Sigma'_w = A^{-1} = \Sigma_w - \frac{\Sigma_w (c z)\, z^\top \Sigma_w}{1 + z^\top \Sigma_w (c z)} = \Sigma_w - \frac{c}{1 + c\, z^\top \Sigma_w z} (\Sigma_w z)(\Sigma_w z)^\top,$$

which is the stated covariance. From Step 2, $\mu'_w = \mu_w + A^{-1}(y - p) z = \mu_w + \Sigma'_w (y - p) z$, since $\Sigma'_w = A^{-1}$. This is the stated mean update. □

B.2 Proof of Lemma 3.2

Proof. By Lemma 3.1, $\Sigma_{\mathcal{T}\cup\{i\}} = \Sigma_{\mathcal{T}} - \frac{(\Sigma_{\mathcal{T}}\zeta_i)(\Sigma_{\mathcal{T}}\zeta_i)^\top}{1+\zeta_i^\top\Sigma_{\mathcal{T}}\zeta_i}$. Taking $-\mathrm{tr}(\cdot\, M)$ of both sides, using linearity of trace and $\mathrm{tr}\big((\Sigma_{\mathcal{T}}\zeta_i)(\Sigma_{\mathcal{T}}\zeta_i)^\top M\big) = \zeta_i^\top\Sigma_{\mathcal{T}}M\Sigma_{\mathcal{T}}\zeta_i$ gives the first equality; expanding M and substituting $\zeta_i = \sqrt{c_i}z_i$ gives the second. Non-negativity follows from $M \succeq 0$. The posterior mean update follows from Lemma 3.1. □

B.3 Proof of Lemma 3.3

Proof. Write $p_i = p_{\mathrm{ans}}(\Delta_i)$. Since i is answered independently of $\mathcal{T}$ with probability p_i, the answered subset of $\mathcal{T}\cup\{i\}$ is $\mathcal{T}_0\cup\{i\}$ with probability p_i and $\mathcal{T}_0$ with probability $1-p_i$, where $\mathcal{T}_0 \subseteq \mathcal{T}$ is the answered subset of $\mathcal{T}$. Hence

$$\begin{aligned} f_{\mathrm{ans}}(\mathcal{T}\cup\{i\}) &= \mathbb{E}\big[p_i\, f(\mathcal{T}_0\cup\{i\}) + (1-p_i)\, f(\mathcal{T}_0)\big] \\ &= f_{\mathrm{ans}}(\mathcal{T}) + p_i\,\mathbb{E}\big[f(\mathcal{T}_0\cup\{i\}) - f(\mathcal{T}_0)\big] \\ &\implies f_{\mathrm{ans}}(\mathcal{T}\cup\{i\}) - f_{\mathrm{ans}}(\mathcal{T}) = p_i\mathbb{E}\big[u_i(\mathcal{T}_0)\big], \end{aligned}$$

which gives the claim. □

B.4 Submodularity

Definition B.1 (Submodular function)**.** *A set function $F : 2^{\mathcal{C}} \to \mathbb{R}$ is* submodular *if for all $\mathcal{L} \subseteq \mathcal{T} \subseteq \mathcal{C}$ and $i \in \mathcal{C}\setminus\mathcal{L}$,*

$$F(\mathcal{L}\cup\{i\}) - F(\mathcal{L}) \;\geq\; F(\mathcal{T}\cup\{i\}) - F(\mathcal{T}).$$

The objective f in Eq. (6) is monotone since the marginal gain $u_i(\mathcal{T}) \geq 0$ by Lemma 3.2), but it is not submodular. It is instead *approximately* submodular, with the approximation captured by the submodularity ratio (Das & Kempe, 2011).

Definition B.2 (Submodularity ratio)**.** *For a ground set $\mathcal{C}$ and $b \in \mathbb{N}$, the submodularity ratio of $F : 2^{\mathcal{C}} \to \mathbb{R}$ is*

$$\gamma_b(F) \;=\; \min_{\mathcal{T},\mathcal{L}\subseteq\mathcal{C}:\ |\mathcal{L}|\leq b,\ \mathcal{L}\cap\mathcal{T}=\emptyset} \frac{\sum_{i\in\mathcal{L}}\big(F(\mathcal{T}\cup\{i\}) - F(\mathcal{T})\big)}{F(\mathcal{T}\cup\mathcal{L}) - F(\mathcal{T})}.$$

The ratio quantifies how much the value of F can increase by adding a whole set $\mathcal{L}$ at once, relative to the sum of the individual gains, that is, how nearly F obeys diminishing returns. If F is submodular, then $\gamma_b(F) = 1$; when $\gamma_b(F) < 1$, the function is approximately submodular.

To bound $\gamma_b(f)$, we exploit the fact that f is a non-negative combination of single-target objectives: writing $f_{\tilde{\phi}}(\mathcal{T}) = -\tilde{\phi}^\top\Sigma_{\mathcal{T}}\tilde{\phi}$, Eq. (6) reads $f = \frac{1}{2n}\sum_{j=1}^{2n} f_{\tilde{\phi}_j}$.

Lemma B.1 (Ratio of a conic combination)**.** *Let $F = \sum_j \alpha_j F_j$ with $\alpha_j \geq 0$ and each F_j monotone. Then $\gamma_b(F) \geq \min_j \gamma_b(F_j)$.*

Proof. Fix $\mathcal{T}, \mathcal{L}$. Monotonicity gives $F_j(\mathcal{T}\cup\mathcal{L}) - F_j(\mathcal{T}) \geq 0$ for every j, and by definition

$$\sum_{i\in\mathcal{L}}\big(F_j(\mathcal{T}\cup\{i\}) - F_j(\mathcal{T})\big) \geq \gamma_b(F_j)\big(F_j(\mathcal{T}\cup\mathcal{L}) - F_j(\mathcal{T})\big).$$

Summing against the weights α_j and lower-bounding each $\gamma_b(F_j)$ by $\min_{j'}\gamma_b(F_{j'})$ gives

$$\sum_{i\in\mathcal{L}}\big(F(\mathcal{T}\cup\{i\}) - F(\mathcal{T})\big) \geq \big(\min_{j'}\gamma_b(F_{j'})\big)\big(F(\mathcal{T}\cup\mathcal{L}) - F(\mathcal{T})\big),$$

which gives the claim. □

Lemma B.2 (Submodularity ratio of f)**.** *For any $b > 0$,*

$$\gamma_b(f) \geq \Big(1 + \max_i \zeta_i^\top\Sigma_w\zeta_i\Big)^{-1} \;\geq\; \Big(1 + \max_i \|z_i\|^2/(4\lambda)\Big)^{-1}.$$

Proof. Fix a target $\widetilde{\phi} \in \{\widetilde{\phi}_1, \ldots, \widetilde{\phi}_{2n}\}$, and let $\mathcal{L} \subseteq \mathcal{C} \setminus \mathcal{T}$ with $|\mathcal{L}| \le b$. Write $v = \Sigma_{\mathcal{T}}\widetilde{\phi}$, stack the variance-scaled contrasts $\zeta_i^\top$ $(i \in \mathcal{L})$ row-wise into $Z_{\mathcal{L}}$, and set $a = Z_{\mathcal{L}}v$, $G = Z_{\mathcal{L}}\Sigma_{\mathcal{T}}Z_{\mathcal{L}}^\top$, and $D = I + \mathrm{diag}(G)$, so that $D_{ii} = 1 + \zeta_i^\top \Sigma_{\mathcal{T}}\zeta_i$. By the Woodbury identity the joint gain is $f_{\widetilde{\phi}}(\mathcal{T} \cup \mathcal{L}) - f_{\widetilde{\phi}}(\mathcal{T}) = a^\top (I+G)^{-1}a$, and the individual gains are $\Delta_i = f_{\widetilde{\phi}}(\mathcal{T} \cup \{i\}) - f_{\widetilde{\phi}}(\mathcal{T}) = a_i^2/D_{ii}$.

Let $\tilde{a} = D^{-1/2}a$. Then $a^\top(I+G)^{-1}a = \tilde{a}^\top\big(D^{-1/2}(I+G)D^{-1/2}\big)^{-1}\tilde{a}$. Since $G \succeq 0$ gives $I + G \succeq I$, we have $D^{-1/2}(I+G)D^{-1/2} \succeq D^{-1}$, hence its inverse is $\preceq D$, and

$$a^\top(I+G)^{-1}a \;\le\; \tilde{a}^\top D\,\tilde{a} \;\le\; \max_i D_{ii}\,\|\tilde{a}\|^2 \;=\; \max_i D_{ii} \sum_{i\in\mathcal{L}} \Delta_i,$$

using $\|\tilde{a}\|^2 = \sum_i a_i^2/D_{ii} = \sum_i \Delta_i$. Thus the ratio of $\sum_i \Delta_i$ to the joint gain is at least $1/\max_i D_{ii}$. Finally, labeling only shrinks the posterior, so $\Sigma_{\mathcal{T}} \preceq \Sigma_w \preceq \lambda^{-1}I$, and $\zeta_i^\top\Sigma_{\mathcal{T}}\zeta_i = c_i\, z_i^\top \Sigma_{\mathcal{T}} z_i \le \|z_i\|^2/(4\lambda)$ since $c_i \le 1/4$. Hence $\max_i D_{ii} \le 1 + \max_i \|z_i\|^2/(4\lambda)$, giving $\gamma_b(f_{\widetilde{\phi}}) \ge (1 + \max_i \|z_i\|^2/(4\lambda))^{-1}$. The bound is uniform in $\widetilde{\phi}_j$, so by Lemma B.1 it transfers to $f = \frac{1}{2n}\sum_j f_{\widetilde{\phi}_j}$. □

Lemma B.3 (Submodularity ratio of f_{ans})**.** *For any answerability probabilities $\{p_{\mathrm{ans}}(\Delta_i)\}_{i=1}^n$ and any $b \in \mathbb{N}$, the weighted objective f_{ans} is monotone and $\gamma_b(f_{\mathrm{ans}}) \ge \gamma_b(f)$.*

Proof. Monotonicity is immediate from Lemma 3.3 and the monotonicity of f. For bounding the submodularity ratio, fix disjoint sets $\mathcal{T}, \mathcal{L}$ with $|\mathcal{L}| \le b$, and let $\mathcal{T}_0 \subseteq \mathcal{T}$, $\mathcal{L}_0 \subseteq \mathcal{L}$ be the corresponding randomly answered subsets. Let $p_i = p_{\mathrm{ans}}(\Delta_i)$. By definition of f_{ans} and by Lemma 3.3,

$$f_{\mathrm{ans}}(\mathcal{T} \cup \mathcal{L}) - f_{\mathrm{ans}}(\mathcal{T}) = \mathbb{E}\big[f(\mathcal{T}_0 \cup \mathcal{L}_0) - f(\mathcal{T}_0)\big],$$
$$\sum\nolimits_{i\in\mathcal{L}} \big(f_{\mathrm{ans}}(\mathcal{T} \cup \{i\}) - f_{\mathrm{ans}}(\mathcal{T})\big) = \mathbb{E}\Big[\sum\nolimits_{i\in\mathcal{L}} p_i\, u_i(\mathcal{T}_0)\Big].$$

For each realization $\mathcal{T}_0$ and $\mathcal{L}_0$, the definition of $\gamma_b(f)$ applied to the pair $(\mathcal{T}_0, \mathcal{L}_0)$ gives

$$f(\mathcal{T}_0 \cup \mathcal{L}_0) - f(\mathcal{T}_0) \le \gamma_b(f)^{-1} \sum\nolimits_{i\in\mathcal{L}_0} u_i(\mathcal{T}_0).$$

Taking expectation over $\mathcal{L}_0$ and using $\Pr[i \in \mathcal{L}_0] = p_i$ yields

$$\mathbb{E}[f(\mathcal{T}_0 \cup \mathcal{L}_0) - f(\mathcal{T}_0) \mid \mathcal{T}_0] \le \gamma_b(f)^{-1} \sum\nolimits_{i\in\mathcal{L}} p_i\, u_i(\mathcal{T}_0).$$

Taking expectation over $\mathcal{T}_0$, we get

$$\begin{aligned}
&\mathbb{E}[f(\mathcal{T}_0 \cup \mathcal{L}_0) - f(\mathcal{T}_0)] \le \gamma_b(f)^{-1}\mathbb{E}\Big[\sum\nolimits_{i\in\mathcal{L}} p_i\, u_i(\mathcal{T}_0)\Big] \\
&\quad\Longrightarrow\; f_{\mathrm{ans}}(\mathcal{T} \cup \mathcal{L}) - f_{\mathrm{ans}}(\mathcal{T}) \le \gamma_b(f)^{-1} \sum\nolimits_{i\in\mathcal{L}} \big(f_{\mathrm{ans}}(\mathcal{T} \cup \{i\}) - f_{\mathrm{ans}}(\mathcal{T})\big) \\
&\quad\Longrightarrow\; \gamma_b(f) \le \frac{\sum_{i\in\mathcal{L}} \big(f_{\mathrm{ans}}(\mathcal{T} \cup \{i\}) - f_{\mathrm{ans}}(\mathcal{T})\big)}{f_{\mathrm{ans}}(\mathcal{T} \cup \mathcal{L}) - f_{\mathrm{ans}}(\mathcal{T})}\,.
\end{aligned}$$

Taking the minimum over all such $\mathcal{T}, \mathcal{L}$ yields the bound. □

B.4.1 Greedy Guarantee

The objective f_{ans} is well suited to greedy selection: it is monotone and, while not submodular, close enough that the greedy solution is provably near-optimal. We measure the gap by the submodularity ratio $\gamma_b(\cdot) \in (0, 1]$, which quantifies how much the value of a set function can exceed the sum of its elements' individual gains, and is equal to 1 for a submodular function (Das & Kempe, 2011).

Recall $f(\mathcal{T}) = -\mathrm{tr}(\Sigma_{\mathcal{T}}M)$ and its answerability-weighted form $f_{\mathrm{ans}}(\mathcal{T}) = \mathbb{E}[f(\mathcal{T}_0)]$. Both are negative for every $\mathcal{T}$, so the guarantee of Das & Kempe (2011), which requires a normalized non-negative objective, does not apply to them directly. We pass to the normalized objectives

$$g(\mathcal{T}) = f(\mathcal{T}) - f(\emptyset) = \mathrm{tr}\big((\Sigma_w - \Sigma_{\mathcal{T}})M\big), \qquad g_{\mathrm{ans}}(\mathcal{T}) = f_{\mathrm{ans}}(\mathcal{T}) - f_{\mathrm{ans}}(\emptyset),$$

which is the (expected) pool variance reduction from labeling $\mathcal{T}$.

Lemma B.4 (Normalization preserves monotonicity and ratio). *g and g_{ans} are monotone increasing with $g(\emptyset) = g_{\text{ans}}(\emptyset) = 0$ and $g, g_{\text{ans}} \geq 0$, and $\gamma_b(g) = \gamma_b(f)$, $\gamma_b(g_{\text{ans}}) = \gamma_b(f_{\text{ans}})$.*

Proof. For any $\mathcal{S}$ and $i \notin \mathcal{S}$, subtracting a constant leaves marginal gains unchanged:

$$g(\mathcal{S} \cup \{i\}) - g(\mathcal{S}) = f(\mathcal{S} \cup \{i\}) - f(\mathcal{S}),$$

and likewise for g_{ans} and f_{ans}. Monotonicity is a property of marginal gains, so it transfers from f (resp. f_{ans}) directly. The submodularity ratio (Definition B.2) is a ratio of sums of marginal gains, so it too is unchanged by the constant shift, giving $\gamma_b(g) = \gamma_b(f)$ and $\gamma_b(g_{\text{ans}}) = \gamma_b(f_{\text{ans}})$. Finally $g(\emptyset) = 0$ by construction, and $g(\mathcal{T}) = \operatorname{tr}((\Sigma_w - \Sigma_{\mathcal{T}})M) \geq 0$ because labeling only contracts the posterior, $\Sigma_{\mathcal{T}} \preceq \Sigma_w$, so $\Sigma_w - \Sigma_{\mathcal{T}} \succeq 0$ and $M \succeq 0$. For g_{ans}, $g_{\text{ans}}(\mathcal{T}) = \mathbb{E}[f(\mathcal{T}_0)] - f(\emptyset) = \mathbb{E}[g(\mathcal{T}_0)] \geq 0$, using that $\mathcal{T}_0 = \emptyset$ contributes $f(\emptyset)$ under the expectation, and $g_{\text{ans}}(\emptyset) = 0$. □

Theorem B.1 (Greedy guarantee). *Greedily maximizing g_{ans} returns a set $\mathcal{T}$ of size at most b with*

$$g_{\text{ans}}(\mathcal{T}) \;\geq\; \left(1 - e^{-\gamma}\right) g_{\text{ans}}(\mathcal{T}^\star), \qquad \gamma = \frac{1}{1 + \max_{i \in \mathcal{C}} \|z_i\|^2/(4\lambda)},$$

where $\mathcal{T}^\star$ is an optimal set of size at most b.

Proof. By Lemma B.4, g_{ans} is monotone, non-negative, and normalized with $g_{\text{ans}}(\emptyset) = 0$, and has the same submodularity ratio as f_{ans}. By Lemmas B.2 and B.3, $\gamma_b(g_{\text{ans}}) = \gamma_b(f_{\text{ans}}) \geq \gamma_b(f) \geq \gamma$. The bound for greedy maximization of a normalized, monotone set function with submodularity ratio at least γ under a cardinality-b constraint (Das & Kempe, 2011) then gives the stated inequality. Since greedy selection uses the marginal gains u_i, which coincide for f_{ans} and g_{ans}, the greedy set returned is identical whether the objective is written as f_{ans} or g_{ans}. □

The bound does not depend on the batch size b. Setting λ adaptively from the mean squared contrast norm, $\lambda \propto \frac{1}{n}\sum_i \|z_i\|^2$, keeps $\|z_i\|^2/(4\lambda) = O(1)$, so γ stays bounded away from zero irrespective of the feature map's scale. Greedy selection is near-optimal, up to the $(1 - 1/e)$ factor, whenever no single contrast carries too much information relative to the prior scale λ. The guarantee holds for a batch objective with each c_i evaluated at the round's prior mean and abstentions treated as uninformative. Our sequential rule, which refreshes $\mu_{\mathcal{T}}$ after each label, is a heuristic refinement that performs better in practice (see Figure C.1).

C Method Details

This appendix gives the missing specification of the methods: the online fine-tuning and calibration loop, the critic gauge (Algorithm 2) and the convention gauge (Appendix C.1), the query selector (Algorithm 3), and the simulated teacher used in all experiments.

C.1 Gauge estimator definitions

Let n denote the number of warmup transitions, E the number of reward-model ensemble members, and H the segment length. The raw learned reward is $\widehat{r}_\psi = E^{-1}\sum_{j=1}^{E} \widehat{r}_{\psi_j}$. The implied reward is

$$\overline{r}_i = Q(s_i, a_i) - \gamma(1 - d_i)\, V(s_i'),$$

where Q is the frozen critic-ensemble mean, γ the discount, and $1 - d_i$ the stored bootstrap mask; $V(s_i')$ is estimated by $Q(s_i', a_i')$ with one draw $a_i' \sim \pi(\cdot \mid s_i')$ from the frozen pretrained policy.

(μ_r, σ_r) are the empirical mean and standard deviation of $\{\overline{r}_i\}_{i=1}^n$, and $(\widehat{\mu}, \widehat{\sigma})$ those of $\{\widehat{r}_\psi(s_i, a_i)\}_{i=1}^n$; standard deviations use the population convention. Correlations are stabilized: $\operatorname{corr}(x, y) = \operatorname{Cov}(x, y)/(\sigma_x \sigma_y + \epsilon_c)$. Then $\varrho = \operatorname{corr}(\widehat{r}_\psi, \overline{r})$ and $\varrho_j = \operatorname{corr}(\widehat{r}_{\psi_j}, \overline{r})$, and

$$\tau^2 = \frac{1}{E}\operatorname{Var}(\varrho_1, \ldots, \varrho_E), \qquad \omega = \max\left\{0,\, 1 - \frac{\tau^2 + 1/n}{\varrho^2 + \epsilon_c}\right\},$$

where Var is the unbiased sample variance and $\tau^2 = 0$ for $E = 1$.

Let $\mathcal{S}_H$ contain the start indices of all stride-1, length-H windows of the chronologically ordered warmup sample that do not cross a stored terminal, with $n_{\text{starts}} = |\mathcal{S}_H|$. When $n_{\text{starts}} \geq 2H$, the

window sums are $\overline{R}_k = \sum_{t=0}^{H-1} \overline{r}_{k+t}$ and $\widehat{R}_k = E^{-1} \sum_j \widehat{R}_{j,k}$ with $\widehat{R}_{j,k} = \sum_{t=0}^{H-1} \widehat{r}_{\psi_j}(s_{k+t}, a_{k+t})$, for $k \in \mathcal{S}_H$. The window correlations are $\varrho_H = \mathrm{corr}(\widehat{R}, \overline{R})$ and $\varrho_{H,j} = \mathrm{corr}(\widehat{R}_j, \overline{R})$, and $(\sigma_{r,H}, \widehat{\sigma}_H)$ are the standard deviations of $(\overline{R}, \widehat{R})$. When window statistics are unavailable, set $\varrho_H = \varrho$.

Define τ_H^2 from $\{\varrho_{H,j}\}$ by the same rule as τ^2, and

$$n_H = \max\Big\{2, \Big\lfloor n_{\text{starts}}/H \Big\rfloor\Big\}, \qquad \omega_H = \max\left\{0,\, 1 - \frac{\tau_H^2 + 1/n_H}{\varrho_H^2 + \epsilon_c}\right\}.$$

If $\omega = 0$ and $\omega_H > 0$, the scale computation replaces $(\varrho, \sigma_r, \widehat{\sigma}, \omega)$ by $(\varrho_H, \sigma_{r,H}, \widehat{\sigma}_H, \omega_H)$; the offset retains the per-step means. Otherwise, $\omega = 0$ gives $\alpha = 0$ and $\beta = \mu_r$.

The scale and offset are

$$\alpha = \omega\, |\varrho|\, \frac{\sigma_r}{\widehat{\sigma} + \epsilon}\, \mathbf{1}\{\varrho_H > 0\}, \qquad \beta = \mu_r - \alpha\, \widehat{\mu},$$

with $\epsilon = 10^{-8}$ and $\epsilon_c = 10^{-12}$.

Convention Gauge In the privileged setting, where the offline dataset $\mathcal{D}$ and its ground-truth rewards r remain accessible during fine-tuning, the scale α and offset β match the first two moments of the true reward distribution. Setting target dispersion $s^\star \leftarrow \sigma_r$:

$$\alpha = \frac{s^\star}{\hat{\sigma} + \epsilon}, \qquad \beta = \mu_r - \alpha\hat{\mu}. \tag{C.1}$$

C.2 Calibrated Teacher Answerability

When $\mathcal{H}$ contains sufficient samples (≥ 4 instances of both answered and tied outcomes), we fit a 1D logistic regression model using regularized Newton iterations:

$$P_{\text{ans}}(\Delta_i) = \begin{cases} \text{sigmoid}(a + b\hat{\Delta}_i), & \text{if } |\mathcal{H}_{\text{ans}}| \geq 4 \text{ and } |\mathcal{H}_{\text{tie}}| \geq 4, \\ \text{sigmoid}(\hat{\Delta}_i), & \text{otherwise (cold-start fallback),} \end{cases} \tag{C.2}$$

where $\hat{\Delta}_i = (\Delta_i - \mu_\Delta)/\sigma_\Delta$ is the standardized return gap.

C.3 Teacher score per environment

LABEL always applies the tie rule above to a scalar segment score $G(\sigma)$; only the score differs by environment. Segments have length H and are written $\sigma = (s_1, a_1, \ldots, s_H, a_H)$ with true per-step rewards r_t.

Kitchen (`segment_return`). The score is the true segment return,

$$G(\sigma) \;=\; \sum_{t=1}^{H} r_t. \tag{C.3}$$

Under the Kitchen convention, r_t is the cumulative count of completed subtasks at step t, so this sum rewards both how many subtasks finish and how early they finish within the segment.

AntMaze and Adroit (`goal_future_success`). Both families use the same teacher: sparse rewards are converted to a per-step time-to-goal target before segment scoring. Let τ be the success threshold, let $T_{\max}$ be the episode step limit, and for step t of an episode let

$$\nu(t) \;=\; \min\{t' \geq t :\; r_{t'} > \tau\}$$

be the first success at or after t (undefined if the episode never succeeds from t onward). Then

$$g_t \;=\; \begin{cases} T_{\max} - \big(\nu(t) - t\big), & \nu(t) \text{ exists}, \\ 0, & \text{otherwise}, \end{cases} \qquad G(\sigma) \;=\; \sum_{t=1}^{H} g_t. \tag{C.4}$$

Steps closer to an upcoming success score higher, steps from which the episode never succeeds score zero, and the segment score aggregates over the window; two segments that never approach success tie at $G = 0$. Note that $\nu(t)$ is read from the recorded episode and may fall beyond the segment window, so the teacher is episode-informed: it rates a clip by what the trajectory goes on to do, not by the clip alone.

C.4 Implementation Details

We now give the settings used in all experiments. ANCOR and WSRL share a single online learner and differ only in the reward $\tilde{r}$ written to the buffer $\mathcal{B}$. PEBBLE is implemented separately. Unless stated otherwise, every value below is shared across tasks and figures, and none is tuned per figure.

Offline checkpoints. Each task uses one Cal-QL checkpoint $(\bar{\pi}, \bar{Q})$, and all arms and seeds of that task start from it. Table C.1 lists the settings that vary by task. Every checkpoint uses a REDQ ensemble of 10 critics with layer normalization and a tanh-squashed Gaussian actor. It is trained with batch size 256, $\gamma = 0.99$, and learning rates 3×10^{-4} for the critic and 10^{-4} for the actor and the temperature. The Cal-QL lower bound V_μ is estimated by Monte-Carlo returns on $\mathcal{D}$. The checkpoint is trained on the affinely transformed reward $c_{\mathrm{s}}\, r + c_{\mathrm{b}}$, so $\bar{Q}$ is expressed in these units. The teacher and WSRL see the same transformed reward online, whereas ANCOR never observes it.

Table C.1: Offline checkpoints. Network widths are given as hidden layers $\times$ units. For Adroit, $\mathcal{D}$ is the binary-reward expert and behavior-cloning demonstrations of Cal-QL.

Task	Critic	Actor	CQL α	$(c_{\mathrm{s}}, c_{\mathrm{b}})$	Offline steps
kitchen-mixed-v0	3×512	3×512	5	$(1, -4)$	75k
kitchen-partial-v0	3×512	3×512	5	$(1, -4)$	100k
antmaze-large-diverse-v2	4×256	2×256	auto (gap 0.8)	$(10, -5)$	450k
antmaze-large-play-v2	4×256	2×256	auto (gap 0.8)	$(10, -5)$	550k
pen-binary-v0	3×512	2×512	1	$(10, 5)$	20k
door-binary-v0	3×512	2×512	1	$(10, 5)$	150k
halfcheetah-medium-replay-v2	2×256	2×256	5	$(1, 0)$	250k
walker2d-medium-replay-v2	2×256	2×256	5	$(1, 0)$	250k

Online fine-tuning. We initialize the online learner from the checkpoint and train it using only online data, with the hyperparameters in Table C.2. At each environment step after warm-up, we sample 1024 transitions from $\mathcal{B}$ and split them into four minibatches of 256. The critic ensemble takes one gradient step on each minibatch, followed by one step each for the actor and temperature on the full batch. Following WSRL, we omit the entropy term from the target in Eq. (1) and use $r + \gamma(1 - d)\, Q_{\theta_*}(s', a')$, where r is the stored reward ($\tilde{r}$ for ANCOR, the environment reward for WSRL), $a' \sim \pi_\phi(\cdot \mid s')$, and Q_{θ_*} is the minimum over two target critics drawn at random from the ensemble. The entropy temperature η enters only through the actor and temperature losses. On AntMaze and Kitchen, we take the maximum of this target over 10 sampled next actions. ANCOR and WSRL share this learner. During the first N_w steps, the agent acts with $\bar{\pi}$ and takes no gradient steps. In ANCOR, the first feedback session runs at step N_w, so every update uses a buffer labeled with the gauged reward $\tilde{r}$.

Table C.2: Online fine-tuning hyperparameters, shared by all ANCOR variants and WSRL.

Parameter	Value
Critic ensemble / target subset	10 / 2, layer normalization
Critic steps per environment step	4 (minibatch 256)
Actor and temperature steps per environment step	1 (batch 1024)
Replay capacity	2×10^6
Discount γ / target update rate	0.99 / 5×10^{-3}
Learning rates (critic / actor / temperature)	3×10^{-4} / 10^{-4} / 10^{-4}
Target entropy	$-\dim(\mathcal{A})$
Entropy term in the target	none
Max-target backup	AntMaze, Kitchen (10 next actions)
Warm-up N_w	5,000 steps
Online horizon N_{on}	100k (Adroit), 300k (otherwise)
Evaluation	every 2,000 steps, 20 episodes

Reward model. The reward model $\hat{r}_\psi$ is an ensemble of $E = 3$ MLPs. Each has three hidden layers of 256 units, LeakyReLU activations, and a linear output. The reward depends on the state alone, $\hat{r}_\psi(s, a) = \hat{r}_\psi(s)$. All members are trained jointly on $\mathcal{P}$ with Adam (learning rate 3×10^{-4}, minibatch 128), each with its own shuffle of the data. A session runs at most 4 passes over $\mathcal{P}$ (8 on Kitchen) and stops early once the training accuracy exceeds 0.97. The weights carry over from

one session to the next. In Eq. (2), the segment return uses the weights $w_t \propto \gamma_{\mathrm{BT}}^t$ with $\gamma_{\mathrm{BT}} = 0.99$, normalized so that $\sum_t w_t = H$. The likelihood thus follows the critic's discounting without changing the scale. Segments have length $H = 50$. We draw a segment by picking a completed episode of length at least H uniformly at random, then a start index uniformly within it. The teacher abstains only on exact ties, $G(\sigma^1) = G(\sigma^2)$. Once $\hat{r}_\psi$ is trained, we fit the gauge (α, β) (Algorithm 2) and relabel all of $\mathcal{B}$ with $\tilde{r} = \alpha\hat{r}_\psi + \beta$.

Query schedule and budgets. A feedback session is triggered every $N_{\mathrm{int}} = 5{,}000$ steps and selects b pairs from a fresh pool of $n = 10b$ candidate pairs ($n = 50b$ on Kitchen). Table C.3 gives b and the budget $n_{\max}$ for each experiment. Abstentions count toward $n_{\max}$. Once $n_{\max}$ is reached, no further queries are made, and $\hat{r}_\psi$ and (α, β) stay fixed for the rest of the run. With $n_{\max} = 30$ and $b = 2$, this happens after 15 sessions, at step 75k.

Table C.3: Query settings per experiment. b is the number of pairs per session and $n_{\max}$ the total number of queried pairs.

Figure	Tasks	b	$n_{\max}$	Seeds
1, 2, C.1	Kitchen	5	300	10
	AntMaze, locomotion	2	120	10
	Adroit	2	40	10
3	Kitchen	1	60	20
	all other tasks	2	30	20
4a	door-binary	2	40	10
4b	kitchen-mixed	1, 1, 1, 2, 5	15, 30, 60, 120, 300	10
	walker2d	1, 1, 1, 2	15, 30, 60, 120	10

Active selection. The features $\phi_\psi(\sigma) \in \mathbb{R}^{256}$ are the penultimate-layer activations summed over the segment and averaged over ensemble members. Likewise, μ_w is initialized to the ensemble average of the final-layer weights. We set $\lambda = 0.01 \cdot \frac{1}{n}\sum_i \|z_i\|^2$ over the current pool, the adaptive choice of Section 3.2.2. At the start of every session, the belief is rebuilt from $\Sigma_w = \lambda^{-1} I$ by applying Eq. (8) once to each label in $\mathcal{P}$ (lines 3–5 of Algorithm 3), with features computed under the current $\hat{r}_\psi$ and p clipped to $[10^{-6}, 1 - 10^{-6}]$. The return gap Δ in Eq. (9) uses the mean of the frozen critic ensemble. Within a segment, $V(s'_k)$ is approximated by $Q(s_{k+1}, a_{k+1})$, and the final step reuses $Q(s_H, a_H)$. The answerability model p_{ans} (Eq. (C.2)) is refit once per session. It uses 60 damped Newton steps with an ℓ_2 penalty of 0.1 on (a, b), applied once $\mathcal{H}$ contains at least four outcomes and both kinds are present. The pool, the features, and p_{ans} are held fixed within a session; only (μ_T, Σ_T) is updated after each label. The batch variant of Figure C.1 scores the pool once under the session-start belief and queries the top b pairs. The uniform arm of Figure 3 draws b pairs at random. The budget study of Figures 4b uses the batch variant, which Figure C.1 shows performs comparably in practice.

Critic gauge. The gauge is refit after every reward-model update on the first N_w transitions of $\mathcal{B}$. These are collected by $\bar{\pi}$ before any gradient step, so the regression is performed on the checkpoint's own state distribution. The sample is the same for the whole run. $\bar{Q}$ is the mean over the 10 frozen critics, and $\bar{V}(s')$ is estimated from a single action $a' \sim \bar{\pi}(\cdot \mid s')$ (Appendix C.1). Only true terminals, not time-limit truncations, delimit the windows in $\mathcal{S}_H$. The convention gauge (Figure 2) computes (μ_r, σ_r) and $(\hat{\mu}, \hat{\sigma})$ over the full offline dataset $\mathcal{D}$ with $\epsilon = 10^{-6}$. The no-gauge arm uses $\tilde{r} = \hat{r}_\psi$.

Teacher. Kitchen and locomotion use the segment-return score (Eq. (C.3)), and AntMaze and Adroit use goal future success (Eq. (C.4)). The threshold τ lies midway between the transformed failure and success rewards, which gives $\tau = 0$ in both families, and $T_{\max}$ is the episode step limit. Since $\nu(t)$ is read from the recorded episode, labels are formed only from completed episodes.

Baselines. WSRL shares the checkpoint, the online learner, the warm-up, and all hyperparameters in Table C.2. Its buffer stores the transformed environment reward, and it trains no reward model and issues no queries. On AntMaze, we additionally supply WSRL with a dense distance penalty, storing $c_{\mathrm{s}}\, r + c_{\mathrm{b}} - C\, d(s)$, where $d(s)$ is the geodesic distance from the current state to the goal in maze cells and $C = 0.2778$. This was the strongest reward-supervised variant we found, which makes the skyline stronger. On the remaining tasks, WSRL uses the environment reward alone.

PEBBLE starts without a checkpoint. It first collects 1,000 random steps and then trains SAC for 4,000 steps on a k-nearest-neighbor state-entropy reward with $k = 5$. When feedback begins, the

critic is re-initialized and the actor is kept. From then on, PEBBLE uses the same N_{int}, b, $n_{\max}$, teacher, H, and N_{on} as ANCOR. It queries the pairs of highest preference entropy, averaged over reward members, from a pool of $10b$ pairs. Its reward model has three members, each a 3×256 LeakyReLU network with a tanh output on state–action inputs. The reward model is trained for at most 200 epochs per session. SAC uses 2×256 networks, batch size 512, one update per environment step, a learning rate of 3×10^{-4}, target rate 0.005, and initial temperature 0.1.

Offline prior (Figure 4a). All three door-binary checkpoints share one architecture: a 2×256 actor and critic with layer normalization, a tanh-squashed Gaussian actor, and a 10-critic ensemble. We pretrain each for 150k steps with CQL ($\alpha = 1$), IQL (expectile 0.7, temperature 0.5), or AWAC. AWAC keeps the IQL networks but fits $V(s)$ to $\mathbb{E}_{a\sim\pi}[Q(s,a)]$. For each algorithm, we evaluate the checkpoints at 50k, 75k, 100k, 125k, and 150k steps, using the zero-shot success rate over 100 episodes, and keep the best one. This gives CQL at 100k (success rate 0.69), IQL at 150k (0.54), and AWAC at 125k (0.53). Within each panel, ANCOR and WSRL fine-tune the same checkpoint with the learner of Table C.2, instantiated at the 2×256 width. All other settings follow door-binary in Figure 1.

Evaluation. The policy is evaluated with its mean action on a separate environment instance. We report D4RL-normalized scores; on Kitchen, the score is the number of completed subtasks at the end of the episode divided by four. On Adroit, we report the fraction of episodes that reach the goal. Curves show the mean and one standard error across seeds at each evaluation step, without smoothing. We use seeds 1–10, and 1–20 for Figure 3.

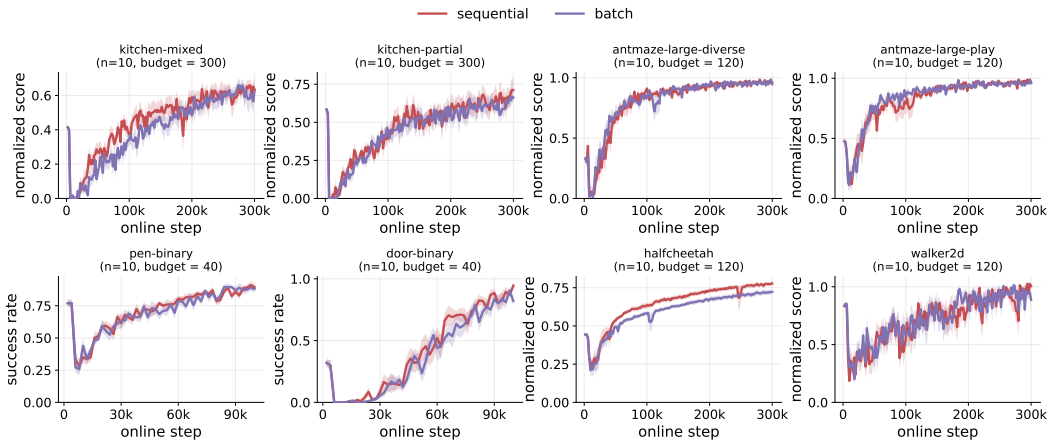


Figure C.1: Effect of sequential acquisition vs batch acquisition. Mean ± one standard error over 10 seeds.

D Other Results

Sequential vs. batch acquisition. Figure C.1 compares ANCOR's performance with sequential acquisition, which updates the posterior after each label, as opposed to batch acquisition, which scores the pool once per session and queries the top b pairs. We observe that the two variants perform on par across all eight tasks.