ACTIVE PREFERENCE QUERYING FOR OFFLINE-TO-ONLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline-to-online reinforcement learning usually assumes scalar rewards during online fine-tuning. We study offline-to-online preference-based RL, where an agent starts from offline preference data and can improve online only through a limited budget of pairwise preference queries, so deciding which comparisons to label is central. We propose an active preference-query rule that selects comparisons that are ambiguous to a reward ensemble and informative for the Bradley–Terry reward estimator, and APOLLO, an algorithm that applies this rule to long-horizon control with offline initialization and a flow-based, action-chunked policy. Our analysis gives a finite-sample bound on the preference-prediction error that holds for adaptively chosen queries; the query rule enters only through how well the labeled data cover the policy's comparisons, so its guarantee improves on random querying when the rule improves this coverage. On eight OGBench manipulation tasks, APOLLO outperforms five preference-based and behavioral-cloning baselines. When only the query rule is changed, our rule achieves the best average and worst-case success among six rules.

1 INTRODUCTION

Preference-based reinforcement learning (PbRL) learns from comparisons between trajectory segments instead of scalar rewards, which is useful when rewards are hard to specify and comparisons are easier for humans to provide. A common approach fits a latent reward model to pairwise preferences with the Bradley–Terry (BT) model (Bradley & Terry, 1952) and optimizes a policy against the learned reward (Christiano et al., 2017; Wirth et al., 2017; Lee et al., 2021b; Park et al., 2022).

Offline-to-online RL initializes an agent from a static dataset and then fine-tunes it through online interaction while handling the resulting distribution shift (Kumar et al., 2020; Kostrikov et al., 2022; Lee et al., 2022; Nakamoto et al., 2023); these methods typically assume that scalar rewards are available for newly collected transitions during online fine-tuning. Offline PbRL learns only from a fixed preference dataset and receives no online feedback (Kang et al., 2023; Gao et al., 2024), while online PbRL methods such as PEBBLE, and those benchmarked in B-Pref, start online, possibly after unsupervised pretraining (Lee et al., 2021b;a). We study the setting that combines (1) offline preference initialization, (2) online interaction, (3) no scalar rewards during online interaction, and (4) a limited budget of actively queried preferences. It arises when rewards are unavailable or costly to engineer, as in robotic manipulation or human-AI interaction, but comparisons of logged behavior can be collected. A label compares only two segments and gives no per-transition signal, so which pairs are queried determines how much each label improves the reward model.

The central question studied in this paper is: ***How should an agent query online preference feedback in order to efficiently improve a policy initialized from offline preference data?***

Random querying can waste labels on obvious or redundant comparisons, and uncertainty sampling can repeatedly select similar ambiguous pairs. Our main contribution is an active preference-query rule (Figure 1) that, in each online round, scores candidate pairs z by $U_t(z) = U_t^{\mathrm{ens}}(z) + U_t^{\mathrm{IG}}(z)$ and queries the top K. The ensemble term U_t^{ens} combines the entropy and the disagreement of the reward ensemble's predictions and flags ambiguous or disputed comparisons. The information-gain term U_t^{IG} weights the Fisher-information leverage of a pair, relative to the information already collected, by its preference ambiguity; it favors reward-parameter directions that earlier queries have not covered and so discourages redundant queries.

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.

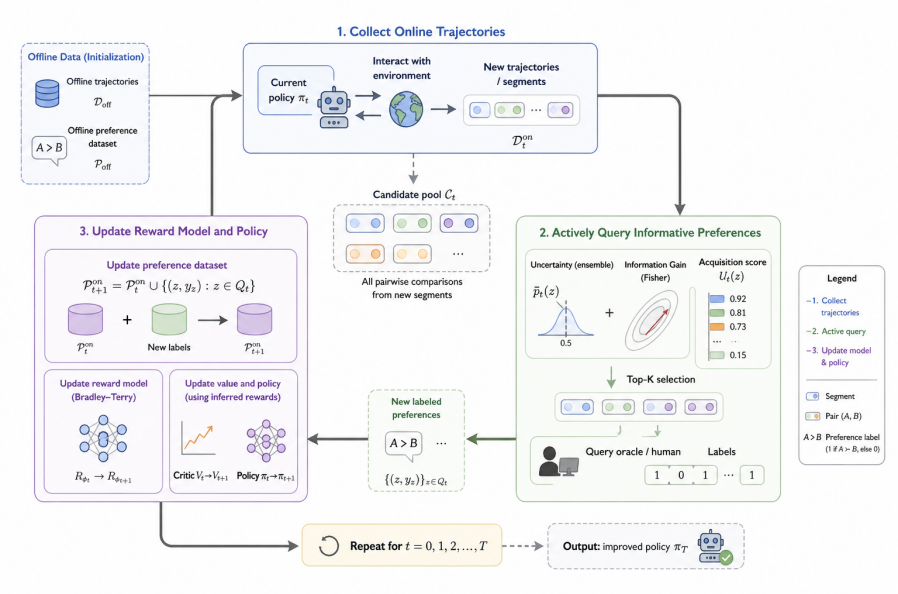

Figure 1: Overview of APOLLO. (1) The current policy collects online segments. (2) The active query rule $U_t = U_t^{\text{ens}} + U_t^{\text{IG}}$ (our main contribution) scores N_c candidate pairs, each pairing a recent on-policy segment with a random segment from the offline–online replay buffer, and queries the top K. (3) The reward ensemble is updated on offline and online preferences, and the offline-initialized, flow-based, action-chunked actor-critic is then updated on the relabeled replay buffer. The pool and oracle boxes are schematic; in our experiments all labels are simulated. Algorithm 1 (Appendix B.4) gives the full loop.

The query rule decides which comparisons are labeled, but it cannot create informative behavior. On our sparse, long-horizon tasks, online learning from scratch has near-zero success within the query budget (Figure 5 in Appendix D), and a Gaussian actor with the same query rule has zero success on DOUBLE-TASK4 and TRIPLE-TASK4 (Appendix D.5). APOLLO therefore couples the rule with offline initialization and an expressive flow-based, action-chunked actor (Li et al., 2025; Lipman et al., 2023). Conversely, in PEBBLE, where unsupervised pretraining provides behavioral coverage, the score has the highest mean on the two tasks where query selection matters, with margins over disagreement sampling within the error bars (Appendix D.6). Our contributions are:

(i) **Active query rule.** For the setting above, we propose the score $U_t = U_t^{\text{ens}} + U_t^{\text{IG}}$, which combines ensemble uncertainty with an ambiguity-weighted Fisher information-gain term to select which online comparisons are labeled (Section 4).

(ii) **Analysis.** The reward model is trained on n_{off} offline and n_{on} actively queried comparisons but is used on comparisons induced by the online policy. For adaptively selected queries, Theorem 1 bounds the preference-prediction error under this online comparison distribution ρ_{on} by $\tilde{O}\big(C_{\text{cov}}(U)\sqrt{(d_{\text{eff}} + \log(1/\delta))/(n_{\text{off}} + n_{\text{on}})}\big)$ with probability $1-\delta$, where d_{eff} is the effective dimension of the reward class and the coverage constant $C_{\text{cov}}(U)$ measures the mismatch between ρ_{on} and the combined training distribution. The query rule enters only through this constant, so the bound improves over random querying by a factor α if $C_{\text{cov}}(U) \le \alpha\, C_{\text{cov}}(\text{rand})$ with $\alpha < 1$. We do not prove that the rule always satisfies this condition; we give a worked example in which it holds, a case in which it fails, and an empirical proxy on two tasks (Appendices D.7 and D.8). Under additional occupancy-transfer and optimization assumptions, Corollary 2 bounds policy suboptimality at rate $(n_{\text{off}} + n_{\text{on}})^{-1/4}$, up to the optimization error.

(iii) **APOLLO** (**A**ctive **P**reference **O**ffline-to-on**L**ine **L**earning and **O**ptimization) instantiates the rule for long-horizon control, combining offline initialization of a BT reward ensemble and policy with a flow-based, action-chunked actor-critic (Section 6).

(iv) **Experiments** (Section 7). On eight OGBench cube-manipulation tasks, the full APOLLO pipeline outperforms five behavioral-cloning and preference-learning baselines. In controlled comparisons that change only the acquisition rule, our score has the highest mean and worst-task success among six rules on four tasks, though it is not the best on every task. With a Gaussian policy backbone and inside PEBBLE, it also attains the highest mean on the tasks where the query rule matters, although several of these margins are within the error bars (Appendices D.4–D.6).

2 Related Work

Online preference-based RL. Building on BT reward learning (Christiano et al., 2017; Wirth et al., 2017), online PbRL methods improve feedback efficiency by combining preference learning with off-policy RL, relabeling, unsupervised pretraining, semi-supervised learning, or data augmentation (Lee et al., 2021b; Park et al., 2022), and B-Pref benchmarks them (Lee et al., 2021a). These methods learn the reward online throughout training, typically from scratch or, as in PEBBLE, after unsupervised pretraining. We instead start from a reward model and policy trained on offline preferences and refine the reward with a limited online query budget.

Offline preference-based RL. Offline PbRL learns from a fixed dataset of transitions and never interacts with the environment. OPPO optimizes policies directly from offline trajectories and preferences without learning a scalar reward model (Kang et al., 2023), while other methods improve the use of trajectory-level preferences through hindsight relabeling, uncertainty-aware alignment, or query-efficient preference selection within a fixed dataset (Gao et al., 2024; Xu et al., 2025b; Yang et al., 2026). These methods use no online feedback; we initialize from offline preferences and then query labels online.

Offline-to-online RL. Offline RL methods such as CQL and IQL address extrapolation error in fixed datasets (Kumar et al., 2020; Kostrikov et al., 2022), and offline-to-online methods stabilize online fine-tuning with pessimistic value estimation, balanced replay, or calibrated pretraining (Lee et al., 2022; Nakamoto et al., 2023). These methods learn from scalar rewards, and the offline-to-online ones also observe them for transitions collected during fine-tuning. We study the combination of offline preference initialization, online interaction, no scalar online reward, and a limited active query budget, in which only a few queried comparisons can correct the offline reward model on the evolving policy's segments.

Active querying, uncertainty, and information gain. Active learning reduces labeling cost by selecting informative examples (Settles, 2009), and has been studied in preference learning for discrete choice models, reward learning, and dueling bandits (Brochu et al., 2007; Bıyık & Sadigh, 2018; Yue et al., 2012; Dudík et al., 2015). Common criteria include entropy, model disagreement, and information gain (MacKay, 1992), and ensembles are a practical proxy for epistemic uncertainty in RL and PbRL (Osband et al., 2016; Lakshminarayanan et al., 2017; Christiano et al., 2017; Lee et al., 2021b; Park et al., 2022; Xu et al., 2025b). Ji et al. (2025) study RLHF with active queries and use an elliptical uncertainty criterion to decide which comparisons to query. We adapt this principle as a D-optimal-type baseline and compare it, together with the entropy and disagreement criteria of PEBBLE and B-Pref, with our score inside the same pipeline (Appendix D.4). Our rule embeds such criteria in an offline-to-online loop and targets the comparison distribution induced by the evolving policy rather than global reward accuracy. If the rule reduces the coverage constant of our analysis relative to random querying, $C_{\mathrm{cov}}(U) \le \alpha\, C_{\mathrm{cov}}(\mathrm{rand})$ with $\alpha < 1$, the preference-error bound improves by the factor α (Section 5). The analysis does not show that the rule always achieves this reduction; Appendices D.7 and D.8 examine when it does, with a worked example, a failure case, and an empirical proxy. **?** study reward-agnostic exploration for PbRL: they collect exploratory trajectories before querying preferences and give guarantees under unknown linear or low-rank transition models. This is complementary to our analysis, which conditions on the policy-induced comparison distribution and gives no exploration guarantee for unknown dynamics.

3 Background

Markov decision processes. We consider a discounted Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $P(s'|s,a)$, reward function $r : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$, discount factor $\gamma \in (0,1)$, and initial-state distribution ρ_0. A policy $\pi(a|s)$ induces trajec-

tories $\tau = (s_0, a_0, s_1, a_1, \ldots)$ and has expected discounted return $J(\pi) = \mathbb{E}_{\tau\sim\pi}\big[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\big]$. In preference-based RL (PbRL) (Christiano et al., 2017; Kang et al., 2023), r is not observed: the learner receives preferences over trajectory segments, fits a reward model to them, and optimizes the policy under the learned reward.

Preferences and the Bradley–Terry model. A segment $\sigma = (s_0, a_0, \ldots, s_{L-1}, a_{L-1})$ has length L, and a comparison is a pair $z = (\sigma^A, \sigma^B)$; $\sigma^A \succ \sigma^B$ means that σ^A is preferred. A reward model $R_\phi : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ induces the segment utility $\bar{R}_\phi(\sigma) = \sum_{(s,a)\in\sigma} R_\phi(s,a)$, and the Bradley–Terry (BT) model (Bradley & Terry, 1952; Wirth et al., 2017) gives the preference probability $p_\phi(z) = P_\phi(\sigma^A \succ \sigma^B) = 1/(1+\exp(-\Delta_\phi(z)))$ with logit $\Delta_\phi(z) = \bar{R}_\phi(\sigma^A) - \bar{R}_\phi(\sigma^B)$. Given a preference dataset $\mathcal{P} = \{(z_i, y_i)\}_{i=1}^n$, where $y_i = 1$ if $\sigma_i^A \succ \sigma_i^B$ and $y_i = 0$ otherwise, the reward model minimizes the negative log-likelihood

$$\mathcal{L}_R(\phi;\mathcal{P}) = -\frac{1}{n}\sum_{i=1}^{n}\big[y_i \log p_\phi(z_i) + (1-y_i)\log\big(1-p_\phi(z_i)\big)\big].$$

Problem setting. We study offline-to-online PbRL defined by four ingredients: (i) *offline preference initialization* from reward-free offline transitions $\mathcal{D}_{\text{off}}$ and a dataset $\mathcal{P}_{\text{off}}$ of n_{off} labeled comparisons; (ii) *online interaction* with the environment; (iii) *no scalar online reward*, so newly collected transitions receive rewards only from the learned reward model; and (iv) a *limited budget* of n_{on} online preference queries, whose labeled comparisons form $\mathcal{P}_{\text{on}}$. The reward model is updated on $\mathcal{P} = \mathcal{P}_{\text{off}} \cup \mathcal{P}_{\text{on}}$. A reward model fitted to $\mathcal{P}_{\text{off}}$ is reliable mainly where $\mathcal{P}_{\text{off}}$ has coverage, but the improving policy visits new regions, so the central question is which online comparisons to label.

4 The Active Preference-Query Rule

This section presents the paper's main methodological contribution: a rule that decides which online comparisons are labeled. It needs only a differentiable reward ensemble, not a particular policy learner. Section 6 instantiates it for long-horizon control in APOLLO; Appendices D.5 and D.6 use it with a Gaussian actor and inside PEBBLE, a purely online PbRL method.

Offline initialization. We maintain an ensemble of reward models $\{R_{\phi_m}\}_{m=1}^M$ whose members have different initializations and data shuffles. Each member is first trained by minimizing $\mathcal{L}_R(\phi_m;\mathcal{P}_{\text{off}})$, and the learned reward relabels $\mathcal{D}_{\text{off}}$ to pretrain the critic and the policy.

Candidate pool. Online training proceeds in rounds $t = 1, 2, \ldots$; in each round the current policy collects new segments, and we form a candidate pool $\mathcal{C}_t = \{z_t^j = (\sigma_{t,j}^A, \sigma_{t,j}^B)\}_{j=1}^{N_c}$: each candidate pairs a recent on-policy segment (the anchor) with a partner segment sampled at random from the combined offline–online replay buffer. Pairing is therefore random, but every candidate contains recent policy behavior; the score U_t below decides which K candidates are labeled. Our experiments use $N_c = 32$, $K = 4$, and $n_{\text{on}} = 1024$.

Ensemble uncertainty. Let $p_{t,m}(z) = p_{\phi_m}(z)$ be the BT probability of the m-th member at round t and $\bar{p}_t(z) = \frac{1}{M}\sum_{m=1}^M p_{t,m}(z)$ the ensemble mean. Following entropy- and disagreement-based acquisition (Settles, 2009; Christiano et al., 2017; Lee et al., 2021b), we define

$$U_t^{\text{ens}}(z) = H_{\text{bin}}(\bar{p}_t(z)) + \text{Std}_{m=1}^M[p_{t,m}(z)], \qquad H_{\text{bin}}(p) = -p\log p - (1-p)\log(1-p).$$

The entropy term peaks at $\bar{p}_t(z) = 0.5$, where the ensemble mean cannot rank the two segments; the standard deviation measures member disagreement, a proxy for epistemic reward uncertainty.

Information gain. Uncertainty alone can repeatedly select similar ambiguous pairs and thus spend labels on redundant comparisons. We therefore add a term from information-based active learning and optimal experimental design (MacKay, 1992; Settles, 2009; Bıyık & Sadigh, 2018). Let $g_{\phi_t}(z) = \nabla_\phi \Delta_{\phi_t}(z) = \nabla_\phi \bar{R}_{\phi_t}(\sigma^A) - \nabla_\phi \bar{R}_{\phi_t}(\sigma^B)$ be the sensitivity of the BT logit to the reward parameters. Under the BT likelihood, labeling z contributes Fisher information proportional to $\bar{p}_t(z)(1-\bar{p}_t(z))g_{\phi_t}(z)g_{\phi_t}(z)^\top$. With the cumulative information matrix of the pairs queried in earlier rounds, $\Lambda_t = \lambda I + \sum_{s<t}\sum_{z'\in\mathcal{Q}_s}\bar{p}_s(z')(1-\bar{p}_s(z'))g_{\phi_s}(z')g_{\phi_s}(z')^\top$, where $\lambda > 0$ ensures invertibility, the information-gain score is

$$U_t^{\text{IG}}(z) = \bar{p}_t(z)(1-\bar{p}_t(z))\, g_{\phi_t}(z)^\top \Lambda_t^{-1} g_{\phi_t}(z).$$

It is large when a pair is ambiguous and its logit gradient points in a reward-parameter direction that earlier queries have not covered, so it discourages redundant comparisons.

Query selection and updates. The active query score and the queried set are

$$U_t(z) = U_t^{\text{ens}}(z) + U_t^{\text{IG}}(z), \qquad \mathcal{Q}_t = \text{TopK}_{z \in \mathcal{C}_t} U_t(z), \qquad |\mathcal{Q}_t| = K,$$

and queries continue until the total online budget n_{on} is spent. The pairs in $\mathcal{Q}_t$ are labeled by a human annotator or a preference oracle and added to the online dataset, $\mathcal{P}_{t+1}^{\text{on}} = \mathcal{P}_t^{\text{on}} \cup \{(z, y_z) : z \in \mathcal{Q}_t\}$ with $\mathcal{P}_1^{\text{on}} = \emptyset$. Each member is then updated by minimizing $\mathcal{L}_R(\phi_m; \mathcal{P}_{t+1})$ on $\mathcal{P}_{t+1} = \mathcal{P}_{\text{off}} \cup \mathcal{P}_{t+1}^{\text{on}}$, the replay buffer is relabeled, and the critic and policy are updated (Algorithm 1, Appendix B.4).

When does active querying help? The score leaves the BT estimator unchanged; it acts only through which pairs are labeled, i.e., through the online query distribution ρ_U, the averaged law of the queried pairs. The support of ρ_U is set by the policy-anchored candidate pool and its allocation within that support by the top-K score, so ρ_U depends on the current policy and the query history and has no closed form. In Section 5, the bound on the preference-prediction error under the online comparison distribution ρ_{on} is proportional to the coverage constant $C_{\text{cov}}(U)$, the largest density ratio of ρ_{on} to the training mixture $\rho^U_{n_{\text{off}}, n_{\text{on}}}$ of offline and queried pairs. When the allocation adds labels where the policy has mass and the offline data are sparse, this constant can fall below that of random querying. If $C_{\text{cov}}(U) \le \alpha\, C_{\text{cov}}(\text{rand})$ with $\alpha < 1$, the bound is a factor α smaller than under random querying; the analysis does not show that the rule always achieves $\alpha < 1$. Appendix D.7 gives a worked example in which it does and a failure case in which it does not, and Appendix D.8 reports a coarse clustered proxy of the constant on two tasks, which is correlational evidence. Appendix B.1 shows a schematic toy illustration of this reallocation.

5 Theoretical Analysis

For the BT reward model trained on n_{off} offline and n_{on} online pairs, the query rule U_t enters the bounds only through a coverage constant. At a fixed constant, offline pairs lower the sufficient online budget; the bounds beat random querying if the rule lowers the constant, which is not proved here.

Setup. For a comparison $z = (\sigma^A, \sigma^B)$, the label $y \in \{0, 1\}$ indicates whether $\sigma^A \succ \sigma^B$; $p^\star(z)$ is the ground-truth BT preference probability induced by an unknown segment utility $\bar{R}^\star$, and $p_\phi(z)$ is that of the reward model R_ϕ. We order the $n = n_{\text{off}} + n_{\text{on}}$ labeled pairs (z_i, y_i) with the offline pairs first, drawn i.i.d. from ρ_{off}, followed by the online pairs in query order (the K pairs of a round in any fixed order). The history $\mathcal{F}_{i-1}$ contains all pairs and labels before z_i and, for an online pair, all randomness used before its round begins (offline training, earlier rollouts, candidate pools, and network updates); online pairs may depend on it arbitrarily. With q_i the conditional law of z_i given $\mathcal{F}_{i-1}$ ($q_i = \rho_{\text{off}}$ offline), the online query distribution is the average $\rho_U = \frac{1}{n_{\text{on}}} \sum_{i > n_{\text{off}}} q_i$, and the training distribution is $\rho^U_{n_{\text{off}}, n_{\text{on}}} = \frac{1}{n} \sum_{i=1}^n q_i = \frac{n_{\text{off}}}{n} \rho_{\text{off}} + \frac{n_{\text{on}}}{n} \rho_U$. Both are data dependent and have no closed form: the support of ρ_U follows the current policy through the on-policy anchors of the candidate pool (Section 4), and the top-K rule allocates labels within that support. The estimator $\hat{\phi}$ minimizes over Φ the empirical BT loss $\widehat{\mathcal{L}}_{n_{\text{off}}, n_{\text{on}}}(\phi) = \frac{1}{n} \sum_{i=1}^n \ell_\phi(z_i, y_i)$, $\ell_\phi(z, y) = -y \log p_\phi(z) - (1 - y) \log(1 - p_\phi(z))$, whose population counterpart is $\mathcal{L}_U(\phi) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\ell_\phi(z_i, y_i) \mid \mathcal{F}_{i-1}] = \mathbb{E}_{z \sim \rho^U_{n_{\text{off}}, n_{\text{on}}},\, y \sim \text{Bern}(p^\star(z))}[\ell_\phi(z, y)]$ under the label model of Assumption 2. The reward model is evaluated on comparisons from ρ_{on}, formed as in the candidate pool from a segment of the final policy and a partner drawn at random from the replay buffer (any ρ_{on} satisfying Assumption 1, e.g., at a checkpoint, can be used).

Assumption 1 (Coverage)**.** *There is a constant $C_{\text{cov}}(U) < \infty$ such that, almost surely, $\sup_{z : \rho^U_{n_{\text{off}}, n_{\text{on}}}(z) > 0} \frac{\rho_{\text{on}}(z)}{\rho^U_{n_{\text{off}}, n_{\text{on}}}(z)} \le C_{\text{cov}}(U)$, and $\rho_{\text{on}}(z) = 0$ whenever $\rho^U_{n_{\text{off}}, n_{\text{on}}}(z) = 0$.*

Assumption 2 (Label model, realizability, and boundedness)**.** *For every i, $y_i \mid \mathcal{F}_{i-1}, z_i \sim \text{Bern}(p^\star(z_i))$. There is $\phi^\star \in \Phi$ such that, almost surely, $p_{\phi^\star} = p^\star$ on the support of $\rho^U_{n_{\text{off}}, n_{\text{on}}}$, and $|\bar{R}_\phi(\sigma)| \le B$ for all $\phi \in \Phi$ and all segments σ.*

Assumption 3 (Finite complexity)**.** *There are constants $d_{\text{eff}} \ge 1$ and $C_0 \ge 1$ such that, for every $\eta \in (0, 1]$, a set $\Phi_\eta \subseteq \Phi$ with $|\Phi_\eta| \le (C_0/\eta)^{d_{\text{eff}}}$ contains, for each $\phi \in \Phi$, some ϕ' with $\sup_z |\Delta_\phi(z) - \Delta_{\phi'}(z)| \le \eta$, where $\Delta_\phi(z) = \bar{R}_\phi(\sigma^A) - \bar{R}_\phi(\sigma^B)$.*

For instance, $d_{\text{eff}} = d$ if $\Phi \subset \mathbb{R}^d$ is bounded and Δ_ϕ is uniformly Lipschitz in ϕ (Appendix A.1).

Proposition 1 (Uniform convergence of the BT likelihood)**.** *Under Assumptions 2–3, with probability at least* $1-\delta$, $\sup_{\phi\in\Phi}\left|\widehat{\mathcal{L}}_{n_{\text{off}},n_{\text{on}}}(\phi)-\mathcal{L}_U(\phi)\right| \le \tilde{O}\left(\sqrt{\frac{d_{\text{eff}}+\log(1/\delta)}{n_{\text{off}}+n_{\text{on}}}}\right)$.

The proof (Appendix A) uses no independence between queries: for fixed ϕ, $\ell_\phi(z_i,y_i)-\mathbb{E}[\ell_\phi(z_i,y_i)\mid \mathcal{F}_{i-1}]$ is a bounded martingale difference sequence, so the Azuma–Hoeffding inequality and a union bound over Φ_η apply. The price of adaptivity is that $\mathcal{L}_U$ and Assumptions 1–2 refer to the data-dependent average of the conditional query laws. By Pinsker's inequality, realizability also yields the required curvature:

Lemma 1 (Curvature of the BT risk)**.** *Under Assumption 2, almost surely, for every* $\phi\in\Phi$, $\mathcal{L}_U(\phi)-\mathcal{L}_U(\phi^\star)=\mathbb{E}_{z\sim\rho^U_{n_{\text{off}},n_{\text{on}}}}\left[\mathrm{KL}\left(\mathrm{Bern}(p^\star(z))\|\mathrm{Bern}(p_\phi(z))\right)\right]\ge 2\,\mathbb{E}_{z\sim\rho^U_{n_{\text{off}},n_{\text{on}}}}\left[|p_\phi(z)-p^\star(z)|^2\right]$.

Theorem 1 (Finite-sample preference convergence)**.** *Under Assumptions 1–3, with probability at least* $1-\delta$*, the empirical BT minimizer* $\hat{\phi}$ *satisfies* $\mathbb{E}_{z\sim\rho_{\text{on}}}\left[|p_{\hat{\phi}}(z)-p^\star(z)|^2\right] \le \tilde{O}\left(C_{\text{cov}}(U)\sqrt{\frac{d_{\text{eff}}+\log(1/\delta)}{n_{\text{off}}+n_{\text{on}}}}\right)$.

What Theorem 1 guarantees. It bounds the squared preference-prediction error under ρ_{on}, and hence, with bounded utilities, the reward-difference error $\mathbb{E}_{\rho_{\text{on}}}[|\Delta_{\hat{\phi}}-\Delta_{\phi^\star}|^2]$ up to a factor κ_B^2 ($\kappa_B = 2+e^{2B}+e^{-2B}$). It does not bound policy suboptimality, which also requires transfer of this error to per-step reward error on the occupancies of an optimal policy $\pi^\star$ and the learned policy $\hat{\pi}$, up to the additive constant that the BT model cannot identify (Assumption 5, with constant C_{occ}), and ϵ_{opt}-optimal policy optimization under the learned reward (Assumption 6). Under these, Corollary 2 (Appendix D.9) gives, for the return $J^\star$ under the ground-truth reward, $J^\star(\pi^\star)-J^\star(\hat{\pi}) \le \tilde{O}\big(\frac{C_{\text{occ}}\kappa_B}{1-\gamma}\big(\frac{C_{\text{cov}}(U)^2(d_{\text{eff}}+\log(1/\delta))}{n_{\text{off}}+n_{\text{on}}}\big)^{1/4}\big)+\epsilon_{\text{opt}}$.

Proposition 2 (Online query complexity)**.** *Under the assumptions of Theorem 1, to guarantee* $\mathbb{E}_{z\sim\rho_{\text{on}}}\left[|p_{\hat{\phi}}(z)-p^\star(z)|^2\right] \le \epsilon^2$ *with probability at least* $1-\delta$*, it suffices to have* $n_{\text{on}} \ge \left[\tilde{O}\left(\frac{C_{\text{cov}}(U)^2(d_{\text{eff}}+\log(1/\delta))}{\epsilon^4}\right)-n_{\text{off}}\right]_+$.

Comparison with random querying. Let $C_{\text{cov}}(\text{rand})$ be the coverage constant, for the same ρ_{on}, when the K pairs of each round are drawn uniformly from identically built candidate pools. If $C_{\text{cov}}(U) \le \alpha C_{\text{cov}}(\text{rand})$ with $\alpha<1$ (Assumption 4, Appendix B.2), the bound of Theorem 1 is at most α times that of random querying; at fixed coverage constants, a total label count N_{rand} sufficient for random querying in Proposition 2 becomes $\alpha^2 N_{\text{rand}}$; and, for a common C_{occ}, the C_{cov}-dependent term of the policy bound shrinks by $\alpha^{1/2}$. These compare upper bounds, and the analysis does not prove that the rule attains $\alpha<1$. The condition requires the rule to place more labels than random querying wherever ρ_{on} exceeds $\alpha C_{\text{cov}}(\text{rand})$ times the random-querying training mixture, i.e., where random querying covers ρ_{on} worst (Appendix A.6). For a concrete two-region example (Appendix D.7), in which Region 1 holds policy-relevant comparisons that are rare offline, take equal offline and online label counts and

$$\rho_{\text{on}}=\left(\tfrac{3}{4},\tfrac{1}{4}\right),\quad \rho_{\text{off}}=\left(\tfrac{1}{4},\tfrac{3}{4}\right),\quad \rho_{\text{rand}}=\left(\tfrac{1}{2},\tfrac{1}{2}\right),\quad \rho_U=(1,0),$$

which give $C_{\text{cov}}(\text{rand})=2$ and $C_{\text{cov}}(U)=1.2$, so $\alpha=0.6$ (and $\alpha\approx 0.85$ at our label split). If instead the policy concentrates on the region the rule does not query, $\rho_{\text{on}}=(\frac{1}{4},\frac{3}{4})$, then $C_{\text{cov}}(U)/C_{\text{cov}}(\text{rand})=5/3$ and the assumption fails. A clustered proxy of the coverage constant (Appendix D.8) is consistent with the condition on TRIPLE-TASK4, where the APOLLO-to-random ratio is 0.85, 0.83, 0.73, and 0.68 after 256, 512, 768, and $1{,}024$ queries and the error bars separate at the full budget, and shows no separation on TRIPLE-TASK1, whose final success is similar under both rules; this evidence is correlational and covers two tasks.

Scope. APOLLO is model-free and forms comparisons from collected trajectories, so the unknown dynamics shape ρ_{on} and ρ_U, but the analysis conditions on them. The ensemble captures preference-model uncertainty, not dynamics uncertainty, and the analysis gives no exploration guarantee for unknown MDPs; **?** study reward-agnostic exploration for PbRL with unknown transitions. The BT label model is also an idealization: noiseless labels that deterministically compare ground-truth returns have $p^\star\in\{0,1\}$, which Assumption 2 excludes.

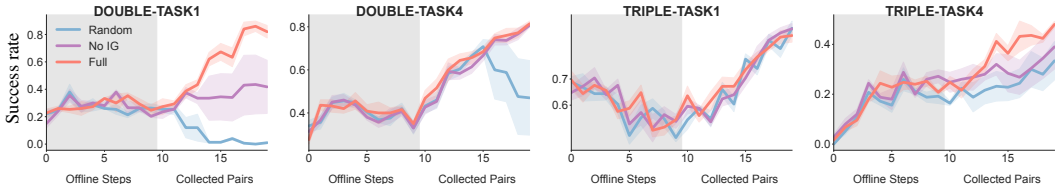

Figure 2: Success rate in $[0, 1]$ for the full score, No-IG (U_t^{ens} only), and random querying; mean and standard error over five seeds. The x-axis spans 1M offline steps (gray; no queries), then up to 1024 collected preference pairs online (white). The TRIPLE-TASK1 drop occurs offline (Appendix D.13).

6 APOLLO: Instantiation for Long-Horizon Control

APOLLO instantiates the query rule of Section 4 for long-horizon control. The rule selects which comparisons are labeled but cannot create informative behavior, so the design choices below target the policy that generates the candidates; Appendix B.5 gives their objectives.

Offline initialization. Learning online from scratch with the same rule and budget has near-zero success on the four tasks tested (Figure 5). The reward ensemble is therefore first trained on $\mathcal{P}_{\text{off}}$, and the critic and actor are pretrained on the relabeled offline transitions.

Action chunking. The actor outputs a chunk $x = (a_0, \ldots, a_{H-1})$ of $H = 5$ actions, executed open-loop and scored by a critic $Q_\psi(s, x)$ (Li et al., 2025). Chunks give temporally coherent behavior and fewer decisions per episode, easing long-horizon credit assignment, and match segment-level preference feedback.

Flow-based actor and distillation. Mixed-quality offline data can contain several behavior modes, which a unimodal Gaussian actor cannot represent, so we use a flow-matching actor (Lipman et al., 2023), as in diffusion and flow policies (Chi et al., 2023; Wang et al., 2023). With a Gaussian Uni-IVR actor (Xu et al., 2025a) instead, success on DOUBLE-TASK4 and TRIPLE-TASK4 is zero under the same query rule (Appendix D.5); that experiment also removes chunking and changes the optimizer. Because multi-step flow sampling is costly online, we execute a one-step actor $\mu_\theta(s, \xi)$, with Gaussian noise ξ, distilled from the flow sampler (Li et al., 2025).

Overall procedure and novelty. Each online round rolls out μ_θ, queries the top $K = 4$ of $N_c = 32$ candidate pairs (Section 4), updates the reward ensemble on $\mathcal{P}_{\text{off}} \cup \mathcal{P}_{\text{on}}$, and updates the critic and actor on the relabeled buffer (Algorithm 1). What is new is the query rule and its integration into an offline-to-online loop without scalar rewards; action chunking, flow policies, and distillation are taken from prior work (Li et al., 2025; Lipman et al., 2023).

7 Experiments

Experimental setting. Each method is trained offline from $\mathcal{P}_{\text{off}}$ (10,000 pairs) and then fine-tuned online; APOLLO queries $n_{\text{on}} = 1024$ labels. We use eight OGBench manipulation tasks (Park et al., 2024) and compare with behavioral cloning of all data (BC-ALL) or of preferred segments (BC-PREF) (Pomerleau, 1991; Ross et al., 2011) and with the preference-based methods IPL (Hejna & Sadigh, 2023), DPO (Rafailov et al., 2023), and DTR (Tu et al., 2025). We report final success after the offline and the online phase, averaged over five seeds as mean $\pm$ standard error. Appendices C.1, C.2, B.5 and Table 4 give datasets, baselines, implementation details, and hyperparameters.

Main comparison. Table 1 compares complete pipelines. APOLLO is best on every task in both phases, whereas the baselines reach at most 0.41 and often fail to improve online; APOLLO improves online on seven of eight tasks (TRIPLE-TASK3 stays at 0.11). The baselines differ from APOLLO in policy class, reward learning, and query selection (IPL and DTR label uniformly sampled online pairs; Appendix C.2), and the offline rows involve no queries; the offline gap thus reflects the reward model and actor, and only the controlled comparisons below isolate the rule.

Table 1: Success rates across eight continuous-control tasks from OGBench. All values are reported as mean ± standard error over five random seeds.

TASK	PHASE	BC-ALL	BC-PREF	IPL	DPO	DTR	APOLLO (OURS)
DOUBLE-TASK1	OFFLINE	0.01±0.02	0.12±0.06	0.05±0.07	0.02±0.03	0.00±0.00	**0.25±0.04**
	ONLINE	0.02±0.03	0.06±0.07	0.07±0.06	0.02±0.03	0.02±0.04	**0.82±0.05**
DOUBLE-TASK2	OFFLINE	0.01±0.02	0.19±0.10	0.01±0.02	0.02±0.03	0.00±0.00	**0.24±0.02**
	ONLINE	0.00±0.00	0.11±0.11	0.10±0.06	0.01±0.02	0.00±0.00	**0.78±0.02**
DOUBLE-TASK3	OFFLINE	0.01±0.02	0.18±0.06	0.05±0.04	0.03±0.03	0.00±0.00	**0.25±0.02**
	ONLINE	0.00±0.00	0.15±0.11	0.07±0.06	0.00±0.00	0.00±0.00	**0.57±0.02**
DOUBLE-TASK4	OFFLINE	0.01±0.02	0.22±0.11	0.06±0.04	0.02±0.03	0.00±0.00	**0.35±0.01**
	ONLINE	0.01±0.02	0.26±0.07	0.11±0.08	0.01±0.02	0.00±0.00	**0.81±0.01**
TRIPLE-TASK1	OFFLINE	0.09±0.04	0.31±0.09	0.13±0.10	0.16±0.16	0.10±0.07	**0.54±0.01**
	ONLINE	0.06±0.06	0.41±0.09	0.11±0.07	0.16±0.07	0.22±0.08	**0.87±0.03**
TRIPLE-TASK2	OFFLINE	0.00±0.00	0.05±0.05	0.00±0.00	0.00±0.00	0.00±0.00	**0.17±0.01**
	ONLINE	0.00±0.00	0.10±0.04	0.03±0.04	0.00±0.00	0.00±0.00	**0.40±0.04**
TRIPLE-TASK3	OFFLINE	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	**0.11±0.02**
	ONLINE	0.00±0.00	0.03±0.05	0.00±0.00	0.00±0.00	0.00±0.00	**0.11±0.01**
TRIPLE-TASK4	OFFLINE	0.00±0.00	0.14±0.07	0.00±0.00	0.00±0.00	0.00±0.00	**0.21±0.01**
	ONLINE	0.00±0.00	0.13±0.10	0.00±0.00	0.00±0.00	0.00±0.00	**0.48±0.01**

Table 2: Final success when only the acquisition rule changes; means over five seeds (standard errors in Table 5). Bold: best in each row. APOLLO denotes the full score U_t ("Full" in Figure 2); all columns run inside the APOLLO pipeline.

TASK	RANDOM	ENTROPY	DISAGREEMENT	D-OPTIMAL	NO-IG	APOLLO
DOUBLE-TASK1	0.01	0.29	0.31	0.37	0.40	**0.82**
DOUBLE-TASK4	0.47	0.30	**0.81**	0.27	**0.81**	**0.81**
TRIPLE-TASK1	0.89	0.86	**0.91**	0.89	0.89	0.87
TRIPLE-TASK4	0.33	0.46	0.37	0.43	0.39	**0.48**
MEAN	0.43	0.48	0.60	0.49	0.62	**0.75**

Isolating the query rule. In Table 2 only the acquisition rule changes; the backbone, reward ensemble, candidate pool, and budget are fixed (Appendix D.4). Entropy and Disagreement are the PEBBLE/B-Pref criteria (Lee et al., 2021b;a); D-optimal adapts the selection principle of Ji et al. (2025) to our pipeline and does not reproduce their full method; No-IG is U_t^{ens} alone. We do not run PEBBLE from scratch, which would conflate the rule with the absence of offline initialization; online-only learning has near-zero success here (Figure 5). The full score has the highest mean (0.75 vs. 0.62 for the next-best rule) and worst-task success (0.48 vs. 0.39), with its largest gain on DOUBLE-TASK1 (0.82 vs. at most 0.40), where random querying falls to near zero within the first half of the online phase (Figure 2). It is not best on every task: TRIPLE-TASK1 is insensitive to the rule (0.86–0.91). A clustered coverage proxy separates the full score from random querying on TRIPLE-TASK4, where the rule changes final success, but not on TRIPLE-TASK1 (Appendix D.8); this evidence is correlational.

Transfer and robustness. With a Gaussian Uni-IVR backbone (Xu et al., 2025a), the score has a higher mean than random querying on all four tasks where that backbone has nonzero success, most clearly on TRIPLE-TASK1 ($0.33 \rightarrow 0.58$) and within about one standard error on DOUBLE-TASK1–2 (Table 3, left; Appendix D.5). Both Gaussian variants fail on DOUBLE-TASK4 and TRIPLE-TASK4, where the flow-based actor succeeds, which is why APOLLO pairs the rule with an expressive actor. In purely online PEBBLE on Meta-World (Yu et al., 2020) and DeepMind Control (Tassa et al., 2018) tasks, where unsupervised pretraining provides the initial behavior, the score has the highest mean on Sweep-Into and Quadruped-Walk, the two tasks where query selection matters, with margins over disagreement sampling within the error bars (Table 3, right; Appendix D.6). Appendices D.10–D.13 vary the information-gain weight, label noise (at 20% flips, three of four tasks keep 94–96% of clean success), the offline/online label split, and the offline training length.

Table 3: Transfer of the query score; means over five seeds (standard errors in Tables 6 and 7). Left: success with a Gaussian Uni-IVR actor, next to the flow-based APOLLO of Table 1. Right: purely online PEBBLE (B-Pref) with uniform, entropy, and disagreement sampling; success rate (%) on Meta-World Button-Press and Sweep-Into, return on DeepMind Control Walker-Walk and Quadruped-Walk. Bold: better Gaussian variant (left) and highest mean (right).

	Gaussian		Flow
Task	Random	APOLLO	APOLLO
double-task1	0.43	**0.51**	0.82
double-task2	0.11	**0.23**	0.78
double-task3	0.15	**0.27**	0.57
double-task4	0.00	0.00	0.81
triple-task1	0.33	**0.58**	0.87
triple-task4	0.00	0.00	0.48

Task	Unif.	Ent.	Dis.	APOLLO
button	100	100	100	100
sweep-into	44.0	53.0	75.0	**78.0**
walker	**895**	879	866	894
quadruped	307	399	510	**622**

8 Conclusion

We studied offline-to-online preference-based RL, where online improvement relies on a limited budget of preference queries. Our main contribution is an active query rule that combines ensemble uncertainty with an ambiguity-weighted information-gain term, together with APOLLO, which applies the rule to long-horizon control. Theoretically, the rule affects the preference-model error only through the coverage of the policy's comparisons, so its guarantee improves on random querying whenever it improves this coverage. Empirically, APOLLO outperforms all baselines on eight OGBench tasks, and with everything else fixed, our rule gives the best mean and worst-task success among six query rules and transfers to a Gaussian policy backbone and to PEBBLE.

9 Limitations

Our preference labels are simulated, and evaluation with human annotators is future work. We do not prove that the rule always improves coverage (Appendices D.7 and D.8 give an example, a failure case, and empirical evidence), and the analysis does not address exploration under unknown dynamics.

AI Use Statement

The research ideas, hypotheses, method, experimental design, implementation, experiments, and interpretation of the results are the work of the authors, who divided the work among themselves, supervised each stage of the project, and independently verified all results and proofs. In this work, we used generative AI tools for the following tasks with required disclosure: assisting, under the authors' direction, with formulating and writing the refined theoretical statements and proofs (Section 5 and Appendix A), every step of which the authors checked. We have not used generative AI tools to generate synthetic data, develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or give feedback on the methodology or experiments, implement methods, or interpret results; translation, cleaning or reformatting datasets, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to search for related literature, to edit and reorganize the manuscript (readability, consistency of notation and reported numbers, LaTeX errors, and moving results from the appendix into the main text), and to render the overview diagram in Figure 1 from the authors' description. All experimental results come from the authors' own runs; no AI tool generated experimental data. No AI tool is an author of this paper. The authors reviewed all AI-assisted work and take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics Statement

This work studies offline-to-online preference-based reinforcement learning and evaluates it only in simulation: robotic manipulation tasks from OGBench (Park et al., 2024). Preference labels

are generated synthetically from ground-truth task returns. The work involves no human subjects, crowdsourcing, or personal data. The offline preference pairs are built from publicly available OGBench simulator data (Appendix C.1), contain no personal data, and pose no foreseeable risk of misuse. We are not aware of direct negative societal impacts of this work.

Reproducibility Statement

All theoretical results are stated with their assumptions in Section 5, Appendix B.2 (comparison with random querying), and Appendix D.9 (policy suboptimality); complete proofs are given in Appendix A and in these appendices. The environments and offline preference datasets are described in Appendix C.1; baseline implementations and our adaptations of them to the online setting are described in Appendix C.2; implementation details and the full algorithm are given in Appendices B.5 and B.4; all hyperparameters are listed in Table 4; and the compute resources and software versions are reported in Appendix C.4. Appendix D describes the additional controlled experiments, including the alternative acquisition rules, the Gaussian Uni-IVR backbone, and the PEBBLE/B-Pref setup (Appendices D.4–D.6). All results are averaged over five random seeds. Anonymized source code, with instructions for reproducing the experiments, will be shared with the reviewers during the discussion period and released upon publication.

References

Erdem Bıyık and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Proceedings of the 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 519–528. PMLR, 2018.

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. doi: 10.1093/biomet/39.3-4.324.

Eric Brochu, Nando de Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Miroslav Dudík, Katja Hofmann, Robert E. Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 563–587. PMLR, 2015.

Chen-Xiao Gao, Shengjun Fang, Chenjun Xiao, Yang Yu, and Zongzhang Zhang. Hindsight preference learning for offline preference-based reinforcement learning. *arXiv preprint arXiv:2407.04451*, 2024.

Joey Hejna and Dorsa Sadigh. Inverse preference learning: Preference-based rl without a reward function. *Advances in Neural Information Processing Systems*, 36:18806–18827, 2023.

Kaixuan Ji, Jiafan He, and Quanquan Gu. Reinforcement learning from human feedback with active queries. *Transactions on Machine Learning Research*, 2025.

Yachen Kang, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. Beyond reward: Offline preference-guided policy optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15753–15768. PMLR, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-Pref: Benchmarking preference-based reinforcement learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021a.

Kimin Lee, Laura M. Smith, and Pieter Abbeel. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6152–6163. PMLR, 2021b.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1702–1712. PMLR, 2022.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. *arXiv preprint arXiv:2507.07969*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. SURF: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations*, 2022.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv:2410.20092*, 2024.

Dean A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991. doi: 10.1162/neco.1991.3.1.88.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Songjun Tu, Jingbo Sun, Qichao Zhang, Yaocheng Zhang, Jia Liu, Ke Chen, and Dongbin Zhao. In-dataset trajectory return regularization for offline preference-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20929–20937, 2025.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2023.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Haoran Xu, Liyuan Mao, Hui Jin, Weinan Zhang, Xianyuan Zhan, and Amy Zhang. Uni-RL: Unifying online and offline RL via implicit value regularization. In *Advances in Neural Information Processing Systems*, 2025a.

Sheng Xu, Bo Yue, Hongyuan Zha, and Guiliang Liu. A distributional approach to uncertainty-aware preference alignment using offline demonstrations. In *International Conference on Learning Representations*, 2025b.

Yiqin Yang, Hao Hu, Yihuan Mao, Jin Zhang, Chengjie Wu, Yuhua Jiang, Xu Yang, Runpeng Xie, Yi Fan, Bo Liu, Yang Gao, Bo Xu, and Chongjie Zhang. OPRIDE: Offline preference-based reinforcement learning via in-dataset exploration. *arXiv preprint arXiv:2604.02349*, 2026.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The K-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012. doi: 10.1016/j.jcss.2011.12.028.

APPENDIX

A MISSING PROOFS

This appendix proves the results of Section 5. Throughout, $n = n_{\text{off}} + n_{\text{on}}$, $\rho^U = \rho^U_{n_{\text{off}},n_{\text{on}}}$, and $\widehat{\mathcal{L}}_n = \widehat{\mathcal{L}}_{n_{\text{off}},n_{\text{on}}}$. The samples (z_i, y_i), the histories $\mathcal{F}_i$, and the conditional query laws q_i are those of the setup in Section 5, with $\mathcal{F}_0$ the history of z_1 (trivial when $n_{\text{off}} > 0$) and $\mathcal{F}_n = \sigma(\mathcal{F}_{n-1}, z_n, y_n)$. By construction, $\mathcal{F}_{i-1} \subseteq \mathcal{F}_i$ and (z_i, y_i) is $\mathcal{F}_i$-measurable, so $(\mathcal{F}_i)_{i=0}^n$ is a filtration to which the sample sequence is adapted. The history $\mathcal{F}_{i-1}$ of an online pair excludes the rollouts and the candidate pool of the round of z_i, beyond the pairs of that round that precede z_i; otherwise z_i, which is a deterministic top-K selection from that pool given the current reward ensemble, would be $\mathcal{F}_{i-1}$-measurable and q_i would be a point mass; the online part of ρ^U would then sit on the finitely many queried pairs and could not help cover any ρ_{on} without atoms. Regular conditional laws q_i exist because comparisons take values in a standard Borel space (pairs of finite sequences of state-action vectors). An expectation $\mathbb{E}_{z\sim\rho}[f(z)]$ integrates over z only; data-dependent quantities inside it, such as $\hat{\phi}$, ρ^U, and ρ_{on}, are held at their realized values. For each realization, we write $\rho(z)$ for the density of a distribution ρ over comparisons with respect to a common reference measure ν (e.g., the sum of all distributions involved), and pointwise relations between densities hold ν-almost everywhere. In particular, Assumption 1 is equivalent to $\rho_{\text{on}}(A) \le C_{\text{cov}}(U)\,\rho^U(A)$ for every measurable set A of comparisons. We take $\delta \in (0,1)$. The notation $\tilde{O}(\cdot)$ hides constants that depend only on B and factors logarithmic in $C_0 n$ (in Proposition 2, logarithmic in C_0, $C_{\text{cov}}(U)$, d_{eff}, and $1/\epsilon$).

A.1 PRELIMINARIES

Let $s(x) = 1/(1+e^{-x})$ and $h(x) = \log(1+e^x)$. For $z = (\sigma^A, \sigma^B)$, $p_\phi(z) = s(\Delta_\phi(z))$ with $\Delta_\phi(z) = \bar{R}_\phi(\sigma^A) - \bar{R}_\phi(\sigma^B)$. Since $-\log s(x) = h(-x)$ and $-\log(1-s(x)) = h(x)$,

$$\ell_\phi(z,y) = y\,h(-\Delta_\phi(z)) + (1-y)\,h(\Delta_\phi(z)), \qquad y \in \{0,1\}.$$

By Assumption 2, $|\bar{R}_\phi(\sigma)| \le B$ for all ϕ and σ, so $|\Delta_\phi(z)| \le 2B$ for all $\phi \in \Phi$ and all z. Three facts follow.

(F1) Boundedness. Since h is positive and increasing,

$$0 < \ell_\phi(z,y) \le h(2B) = \log\left(1+e^{2B}\right) =: C_B,$$

and $c_B \le p_\phi(z) \le 1 - c_B$ with $c_B = 1/(1+e^{2B})$.

(F2) Lipschitz continuity in the logit. Since $h'(x) = s(x) \in (0,1)$, h is 1-Lipschitz, and hence

$$|\ell_\phi(z,y) - \ell_{\phi'}(z,y)| \le |\Delta_\phi(z) - \Delta_{\phi'}(z)| \qquad \text{for all } \phi, \phi' \in \Phi,\ z,\ y.$$

(F3) Compensator. Let $\bar{\ell}_\phi(z) = p^\star(z)\,h(-\Delta_\phi(z)) + (1-p^\star(z))\,h(\Delta_\phi(z))$ be the expected loss at z when $y \sim \text{Bern}(p^\star(z))$. Because $\ell_\phi(z,y)$ is affine in y and, by the label model of Assumption 2, $\mathbb{E}[y_i \mid \mathcal{F}_{i-1}, z_i] = p^\star(z_i)$, the tower property gives $\mathbb{E}[\ell_\phi(z_i,y_i) \mid \mathcal{F}_{i-1}] = \mathbb{E}[\bar{\ell}_\phi(z_i) \mid \mathcal{F}_{i-1}] = \int \bar{\ell}_\phi\, dq_i$. Averaging over i,

$$\mathcal{L}_U(\phi) = \frac{1}{n}\sum_{i=1}^n \int \bar{\ell}_\phi\, dq_i = \mathbb{E}_{z\sim\rho^U}\left[\bar{\ell}_\phi(z)\right],$$

which is the second expression for $\mathcal{L}_U$ in Section 5. We take this right-hand side as the definition of $\mathcal{L}_U(\phi)$ for all $\phi \in \Phi$ simultaneously, so that $\mathcal{L}_U(\hat{\phi})$ is well defined; for each fixed ϕ, it equals $\frac{1}{n}\sum_{i=1}^n \mathbb{E}[\ell_\phi(z_i,y_i) \mid \mathcal{F}_{i-1}]$ almost surely. For the offline pairs, which are i.i.d. draws (z_i, y_i) with $z_i \sim \rho_{\text{off}}$, $q_i = \rho_{\text{off}}$.

A sufficient condition for Assumption 3. Suppose $\Phi \subseteq \{\phi \in \mathbb{R}^d : \|\phi\|_2 \le W\}$ and $\phi \mapsto \bar{R}_\phi(\sigma)$ is $(L_\Delta/2)$-Lipschitz uniformly over segments σ, so that $\sup_z |\Delta_\phi(z) - \Delta_{\phi'}(z)| \le L_\Delta \|\phi - \phi'\|_2$ for all $\phi, \phi' \in \Phi$. This holds, for example, for a reward network with bounded inputs and Lipschitz activations whose d parameters are restricted to a bounded set, since such a network is Lipschitz

in its parameters on that set and segments have a fixed length. Fix $\eta \in (0, 1]$, let $r = \eta/L_\Delta$, and let Φ_η be a maximal r-separated subset of Φ. By maximality, every $\phi \in \Phi$ is within distance r of some $\phi' \in \Phi_\eta$, so $\sup_z |\Delta_\phi(z) - \Delta_{\phi'}(z)| \le \eta$. The Euclidean balls of radius $r/2$ centred at the points of Φ_η are disjoint and lie in the ball of radius $W + r/2$, so comparing volumes gives $|\Phi_\eta| \le (1 + 2W/r)^d = (1 + 2L_\Delta W/\eta)^d \le ((1 + 2L_\Delta W)/\eta)^d$, where the last step uses $\eta \le 1$. Hence Assumption 3 holds with $d_{\text{eff}} = d$ and $C_0 = 1 + 2L_\Delta W$; the Lipschitz constant enters the bounds only through $\log C_0$.

A.2 Proof of Prop. 1

We prove the explicit bound: there is an event $\mathcal{E}$ of probability at least $1 - \delta$ on which

$$\sup_{\phi \in \Phi} \left| \widehat{\mathcal{L}}_n(\phi) - \mathcal{L}_U(\phi) \right| \le \varepsilon_n := C_B \sqrt{\frac{d_{\text{eff}} \log(C_0 n) + \log(2/\delta)}{2n}} + \frac{2}{n}. \tag{1}$$

Proof. Step 1 (a fixed parameter). Fix $\phi \in \Phi$ and define

$$D_i(\phi) = \ell_\phi(z_i, y_i) - \mathbb{E}[\ell_\phi(z_i, y_i) \mid \mathcal{F}_{i-1}], \qquad i = 1, \dots, n,$$

so that, almost surely by (F3), $\widehat{\mathcal{L}}_n(\phi) - \mathcal{L}_U(\phi) = \frac{1}{n} \sum_{i=1}^n D_i(\phi)$. Since (z_i, y_i) is $\mathcal{F}_i$-measurable, $D_i(\phi)$ is $\mathcal{F}_i$-measurable and $\mathbb{E}[D_i(\phi) \mid \mathcal{F}_{i-1}] = 0$, so $(D_i(\phi))_{i \le n}$ is a martingale difference sequence with respect to $(\mathcal{F}_i)$. By (F1), $A_i \le D_i(\phi) \le A_i + C_B$ with $A_i = -\mathbb{E}[\ell_\phi(z_i, y_i) \mid \mathcal{F}_{i-1}]$, which is $\mathcal{F}_{i-1}$-measurable. The Azuma–Hoeffding inequality for martingale differences with predictable ranges of length C_B, applied to $\sum_{i=1}^n D_i(\phi)$ at level nt, gives, for every $t > 0$,

$$\Pr\left(\left| \widehat{\mathcal{L}}_n(\phi) - \mathcal{L}_U(\phi) \right| \ge t \right) \le 2 \exp\left(-\frac{2(nt)^2}{n C_B^2} \right) = 2 \exp\left(-\frac{2nt^2}{C_B^2} \right).$$

The concentration inequality uses neither independence between the samples nor the label model: the online pairs may be selected by any rule, including the query score U_t applied to a candidate pool generated by the current policy. The label model is needed only to identify the compensator $\frac{1}{n} \sum_{i=1}^n \mathbb{E}[\ell_\phi(z_i, y_i) \mid \mathcal{F}_{i-1}]$ with $\mathcal{L}_U(\phi) = \mathbb{E}_{z \sim \rho^U}[\bar{\ell}_\phi(z)]$ in (F3).

Step 2 (union bound over a cover). Let $\eta = 1/n \in (0, 1]$ and let Φ_η be the set of Assumption 3, so $|\Phi_\eta| \le (C_0 n)^{d_{\text{eff}}}$. The set Φ_η is fixed in advance: it is defined through the supremum over all z and does not depend on the samples, so Step 1 applies to each of its elements. With

$$t = C_B \sqrt{\frac{\log(2|\Phi_\eta|/\delta)}{2n}},$$

a union bound over Φ_η shows that the event

$$\mathcal{E} = \left\{ \max_{\phi' \in \Phi_\eta} \left| \widehat{\mathcal{L}}_n(\phi') - \mathcal{L}_U(\phi') \right| < t \right\}$$

is measurable, because Φ_η is finite, and has probability at least $1 - 2|\Phi_\eta| \exp(-2nt^2/C_B^2) = 1 - \delta$.

Step 3 (discretization). Fix $\phi \in \Phi$ and choose $\phi' \in \Phi_\eta$ with $\sup_z |\Delta_\phi(z) - \Delta_{\phi'}(z)| \le \eta$. By (F2), $|\ell_\phi(z, y) - \ell_{\phi'}(z, y)| \le \eta$ for all (z, y). Averaging over the samples gives $|\widehat{\mathcal{L}}_n(\phi) - \widehat{\mathcal{L}}_n(\phi')| \le \eta$. The expected loss $\bar{\ell}_\phi(z)$ of (F3) is a convex combination of $\ell_\phi(z, 1)$ and $\ell_\phi(z, 0)$, so $|\bar{\ell}_\phi(z) - \bar{\ell}_{\phi'}(z)| \le \eta$ for every z, and integrating against ρ^U gives $|\mathcal{L}_U(\phi) - \mathcal{L}_U(\phi')| \le \eta$. Hence, on $\mathcal{E}$,

$$\left| \widehat{\mathcal{L}}_n(\phi) - \mathcal{L}_U(\phi) \right| \le \left| \widehat{\mathcal{L}}_n(\phi') - \mathcal{L}_U(\phi') \right| + 2\eta < t + \frac{2}{n} \qquad \text{for every } \phi \in \Phi.$$

Since $\log(2|\Phi_\eta|/\delta) \le d_{\text{eff}} \log(C_0 n) + \log(2/\delta)$, this proves (1) on $\mathcal{E}$. (The supremum over the possibly uncountable Φ need not be measurable; the statement is that it is bounded on the measurable event $\mathcal{E}$.)

Step 4 (rate). For $n = 1$, the supremum is at most C_B by (F1), and $C_B \le C_B \sqrt{d_{\text{eff}} + \log(1/\delta)}$, which is of the stated form. For $n \ge 2$, because $d_{\text{eff}} \ge 1$, $C_0 \ge 1$, and $\log(2C_0 n) \ge 1$,

$d_{\text{eff}} \log(C_0 n) + \log(2/\delta) \le (d_{\text{eff}} + \log(1/\delta)) \log(2C_0 n)$, and $2/n \le 2\sqrt{(d_{\text{eff}} + \log(1/\delta))/n}$. Therefore

$$\varepsilon_n \le \left(C_B \sqrt{\tfrac{1}{2} \log(2C_0 n)} + 2 \right) \sqrt{\frac{d_{\text{eff}} + \log(1/\delta)}{n_{\text{off}} + n_{\text{on}}}} = \tilde{O}\left(\sqrt{\frac{d_{\text{eff}} + \log(1/\delta)}{n_{\text{off}} + n_{\text{on}}}} \right).$$

□

A.3 Proof of Lemma 1

Proof. Fix a realization on which Assumption 2 holds, and write $H(a) = -a \log a - (1-a)\log(1-a)$ and $\mathrm{kl}(a,b) = \mathrm{KL}(\mathrm{Bern}(a) \| \mathrm{Bern}(b)) = a \log \frac{a}{b} + (1-a) \log \frac{1-a}{1-b}$. For every z and ϕ, the expected loss of (F3) is a cross-entropy, so

$$\bar{\ell}_\phi(z) = -p^\star(z) \log p_\phi(z) - (1 - p^\star(z)) \log(1 - p_\phi(z)) = H(p^\star(z)) + \mathrm{kl}\big(p^\star(z), p_\phi(z)\big).$$

All terms are finite because $p_\phi(z) \in [c_B, 1 - c_B]$ by (F1). By realizability, $p_{\phi^\star} = p^\star$ ρ^U-almost everywhere, so $\bar{\ell}_{\phi^\star}(z) = H(p^\star(z))$ for ρ^U-almost every z. By (F3),

$$\mathcal{L}_U(\phi) - \mathcal{L}_U(\phi^\star) = \mathbb{E}_{z \sim \rho^U} \left[\bar{\ell}_\phi(z) - \bar{\ell}_{\phi^\star}(z) \right] = \mathbb{E}_{z \sim \rho^U} \left[\mathrm{kl}\big(p^\star(z), p_\phi(z)\big) \right].$$

Pinsker's inequality states that $\mathrm{KL}(P \| Q) \ge 2\,\mathrm{TV}(P,Q)^2$, and the total-variation distance between $\mathrm{Bern}(a)$ and $\mathrm{Bern}(b)$ is $\frac{1}{2}(|a-b| + |(1-a) - (1-b)|) = |a-b|$. Hence $\mathrm{kl}(p^\star(z), p_\phi(z)) \ge 2|p_\phi(z) - p^\star(z)|^2$ for every z, and taking expectations under ρ^U proves the lemma. The inequality holds for every $\phi \in \Phi$, not only near $\phi^\star$, so it can be applied to the data-dependent estimator $\hat{\phi}$. □

A.4 Proof of Theorem 1

We prove the explicit form: with probability at least $1 - \delta$,

$$\mathbb{E}_{z \sim \rho^U} \left[|p_{\hat{\phi}}(z) - p^\star(z)|^2 \right] \le \varepsilon_n \qquad \text{and} \qquad \mathbb{E}_{z \sim \rho_{\text{on}}} \left[|p_{\hat{\phi}}(z) - p^\star(z)|^2 \right] \le C_{\text{cov}}(U)\, \varepsilon_n,$$

with ε_n from (1); the stated rate then follows from Step 4 of the proof of Proposition 1.

Proof. Let $\mathcal{E}$ be the event of Proposition 1, which has probability at least $1 - \delta$, intersected with the probability-one events on which Assumptions 1 and 2 hold.

Step 1 (excess risk). On $\mathcal{E}$, using the uniform bound twice and $\widehat{\mathcal{L}}_n(\hat{\phi}) \le \widehat{\mathcal{L}}_n(\phi^\star)$, which holds because $\hat{\phi}$ minimizes $\widehat{\mathcal{L}}_n$ over Φ and $\phi^\star \in \Phi$,

$$\mathcal{L}_U(\hat{\phi}) \le \widehat{\mathcal{L}}_n(\hat{\phi}) + \varepsilon_n \le \widehat{\mathcal{L}}_n(\phi^\star) + \varepsilon_n \le \mathcal{L}_U(\phi^\star) + 2\varepsilon_n.$$

Step 2 (error under the training distribution). Lemma 1 holds simultaneously for all $\phi \in \Phi$, so it applies to $\phi = \hat{\phi}$:

$$2\, \mathbb{E}_{z \sim \rho^U} \left[|p_{\hat{\phi}}(z) - p^\star(z)|^2 \right] \le \mathcal{L}_U(\hat{\phi}) - \mathcal{L}_U(\phi^\star) \le 2\varepsilon_n,$$

which is the first bound.

Step 3 (transfer to ρ_{on}). Assumption 1 gives $\rho_{\text{on}}(z) \le C_{\text{cov}}(U) \rho^U(z)$ for ν-almost every z (both sides vanish where $\rho^U(z) = 0$). For the nonnegative function $f(z) = |p_{\hat{\phi}}(z) - p^\star(z)|^2$, with $\hat{\phi}$ held at its realized value,

$$\mathbb{E}_{z \sim \rho_{\text{on}}}[f(z)] = \int f(z) \rho_{\text{on}}(z)\, d\nu(z) \le C_{\text{cov}}(U) \int f(z) \rho^U(z)\, d\nu(z) \le C_{\text{cov}}(U)\, \varepsilon_n,$$

which is the second bound. □

If $\hat{\phi}$ minimizes $\widehat{\mathcal{L}}_n$ only up to an additive optimization error ϵ_{erm}, Step 1 gives $\mathcal{L}_U(\hat{\phi}) - \mathcal{L}_U(\phi^\star) \le 2\varepsilon_n + \epsilon_{\text{erm}}$, and both bounds hold with ε_n replaced by $\varepsilon_n + \epsilon_{\text{erm}}/2$. The proof uses the coverage condition only in Step 3, and only for the realized distributions, so the result holds for any target distribution ρ_{on}, possibly data dependent, that satisfies Assumption 1. For the same reason, both bounds also hold with $C_{\text{cov}}(U)$ replaced by the realized supremum of ρ_{on}/ρ^U, the smallest constant for the realized distributions, which is the constant used in Appendices A.6 and B.2.

A.5 Proof of Prop. 2

Proof. We may assume $\epsilon \le 1$, since $|p_{\hat{\phi}}(z) - p^\star(z)| \le 1$ makes the claim trivial otherwise. Integrating the inequality of Assumption 1 over z gives $1 \le C_{\mathrm{cov}}(U)$. By the explicit form of Theorem 1, with probability at least $1-\delta$,

$$\mathbb{E}_{z\sim\rho_{\mathrm{on}}}\left[|p_{\hat{\phi}}(z) - p^\star(z)|^2\right] \le C_{\mathrm{cov}}(U)\, C_B \sqrt{\frac{d_{\mathrm{eff}}\log(C_0 n) + \log(2/\delta)}{2n}} + \frac{2\, C_{\mathrm{cov}}(U)}{n}.$$

The right-hand side is at most ϵ^2 if each of the two terms is at most $\epsilon^2/2$. Squaring and rearranging, the first condition is

$$n \ge \frac{2\, C_B^2\, C_{\mathrm{cov}}(U)^2\, (d_{\mathrm{eff}}\log(C_0 n) + \log(2/\delta))}{\epsilon^4},$$

and the second is $n \ge 4C_{\mathrm{cov}}(U)/\epsilon^2$, which is implied by $n \ge 4C_{\mathrm{cov}}(U)^2/\epsilon^4$ because $C_{\mathrm{cov}}(U) \ge 1 \ge \epsilon^2$. Both conditions therefore hold when $n \ge T(n)$, where

$$T(n) = \frac{C_{\mathrm{cov}}(U)^2}{\epsilon^4}\left(2C_B^2\,(d_{\mathrm{eff}}\log(C_0 n) + \log(2/\delta)) + 4\right) = a\log(C_0 n) + b, \tag{2}$$

with $a = 2C_B^2 C_{\mathrm{cov}}(U)^2 d_{\mathrm{eff}}/\epsilon^4$ and $b = C_{\mathrm{cov}}(U)^2\left(2C_B^2\log(2/\delta) + 4\right)/\epsilon^4 \ge 4$. The threshold $T(n)$ depends on n through $\log(C_0 n)$. To remove this dependence, we use an elementary fact: for $c \ge 1$, $m \ge 4c\log(2c)$ implies $m \ge 2c\log m$. Indeed, $m \mapsto m - 2c\log m$ is nondecreasing on $[2c,\infty)$, $4c\log(2c) \ge 4c\log 2 \ge 2c$, and at $m = 4c\log(2c)$ the map equals $2c\log\left(c/\log(2c)\right) \ge 0$, because $u/2 \ge \log u$ for all $u > 0$. Let $\bar{a} = \max\{a, 1\}$. Applying the fact with $m = C_0 n$ and $c = C_0\bar{a}$ shows that $n \ge 4\bar{a}\log(2C_0\bar{a})$ implies $n/2 \ge \bar{a}\log(C_0 n) \ge a\log(C_0 n)$. Hence

$$n \ge 4\bar{a}\log(2C_0\bar{a}) + 2b = \tilde{O}\left(\frac{C_{\mathrm{cov}}(U)^2\,(d_{\mathrm{eff}} + \log(1/\delta))}{\epsilon^4}\right)$$

implies $n \ge a\log(C_0 n) + b = T(n)$. Here $\bar{a} \le a + 1 \le (2C_B^2 + 1)C_{\mathrm{cov}}(U)^2 d_{\mathrm{eff}}/\epsilon^4$, and $\tilde{O}$ hides constants depending on B and the factor $\log(2C_0\bar{a})$, which is logarithmic in C_0, $C_{\mathrm{cov}}(U)$, d_{eff}, and $1/\epsilon$. Since $n = n_{\mathrm{off}} + n_{\mathrm{on}}$ and the number of online queries is nonnegative, it suffices that

$$n_{\mathrm{on}} \ge \left[\tilde{O}\left(\frac{C_{\mathrm{cov}}(U)^2\,(d_{\mathrm{eff}} + \log(1/\delta))}{\epsilon^4}\right) - n_{\mathrm{off}}\right]_+.$$

□

Comparison of sufficient budgets. The comparison with random querying uses the condition $n \ge T(n)$ of (2), whose threshold is exactly proportional to $C_{\mathrm{cov}}(U)^2$; as in the proof, $\epsilon \le 1$. The condition is preserved when n grows: if $n_0 \ge 1/C_0$ and $n_0 \ge T(n_0)$, then $n_0 \ge b \ge 4$, so $\log(C_0 n_0) \ge 1$ and $n_0 \ge a\log(C_0 n_0) \ge a$, and $n \mapsto n - a\log(C_0 n)$ is nondecreasing on $[a,\infty)$; hence $n \ge T(n)$ for every $n \ge n_0$. Treat the coverage constants as fixed across the budgets compared (in general, both depend on n_{off} and n_{on} through the training mixtures), and let T_{rand} and b_{rand} be T and b with $C_{\mathrm{cov}}(\mathrm{rand})$ in place of $C_{\mathrm{cov}}(U)$. If a total $N_{\mathrm{rand}} \ge 1$ satisfies $N_{\mathrm{rand}} \ge T_{\mathrm{rand}}(N_{\mathrm{rand}})$, then $N_{\mathrm{rand}} \ge b_{\mathrm{rand}}$, and under Assumption 4, $b = (C_{\mathrm{cov}}(U)/C_{\mathrm{cov}}(\mathrm{rand}))^2 b_{\mathrm{rand}} \le \alpha^2 b_{\mathrm{rand}}$, so $\alpha^2 N_{\mathrm{rand}} \ge b \ge 4$ and $T_{\mathrm{rand}}(\alpha^2 N_{\mathrm{rand}}) \ge 0$. Because $T = (C_{\mathrm{cov}}(U)/C_{\mathrm{cov}}(\mathrm{rand}))^2 T_{\mathrm{rand}}$, T_{rand} is nondecreasing, and $\alpha^2 N_{\mathrm{rand}} \le N_{\mathrm{rand}}$,

$$T(\alpha^2 N_{\mathrm{rand}}) \le \alpha^2 T_{\mathrm{rand}}(\alpha^2 N_{\mathrm{rand}}) \le \alpha^2 T_{\mathrm{rand}}(N_{\mathrm{rand}}) \le \alpha^2 N_{\mathrm{rand}},$$

so every total $n \ge \alpha^2 N_{\mathrm{rand}}$, i.e., every online budget $n_{\mathrm{on}} \ge [\alpha^2 N_{\mathrm{rand}} - n_{\mathrm{off}}]_+$, suffices for active querying. This gives the comparison of sufficient budgets in Section 5 and Appendix B.2. It compares sufficient conditions and is not a lower bound on the labels that random querying needs.

A.6 An Equivalent Form of the Coverage-Improvement Condition

Let $w = n_{\mathrm{on}}/n$, and take $C_{\mathrm{cov}}(U)$ to be the smallest constant that satisfies the inequality of Assumption 1 for the realized distributions. Since $\rho^U = (1-w)\rho_{\mathrm{off}} + w\rho_U$, the condition $C_{\mathrm{cov}}(U) \le \alpha C_{\mathrm{cov}}(\mathrm{rand})$ of Assumption 4 holds if and only if

$$w\,\rho_U(z) \;\ge\; \frac{\rho_{\mathrm{on}}(z)}{\alpha\, C_{\mathrm{cov}}(\mathrm{rand})} - (1-w)\,\rho_{\mathrm{off}}(z) \qquad \text{for all } z.$$

The online queries must therefore supply at least this much mass wherever the offline data undercover ρ_{on}. In particular, with $\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}} = (1-w)\rho_{\text{off}} + w\rho_{\text{rand}}$ as in Appendix B.2, at every z with $\rho_{\text{on}}(z) > \alpha C_{\text{cov}}(\text{rand})\,\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}}(z)$ the display gives $w\rho_U(z) > \rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}}(z) - (1-w)\rho_{\text{off}}(z) = w\rho_{\text{rand}}(z)$, so the rule must place more mass there than random querying. Because only the online part of the mixture depends on the rule, it can reallocate at most a fraction w of the training mass; in our experiments $n_{\text{off}} = 10{,}000$ and $n_{\text{on}} = 1{,}024$, so $w \approx 0.093$. Appendix D.7 evaluates the worked example at this split.

B Additional Discussions

B.1 Schematic Illustration of Active Querying

Figures 3 and 4 illustrate, on a synthetic two-dimensional comparison space, how a query rule can change the preference training distribution. They are schematic illustrations of the mechanism analyzed in Section 5. The reward model is trained on the combined distribution $\rho^{U}_{n_{\text{off}},n_{\text{on}}}$ of Section 5, which mixes ρ_{off} and the online query distribution ρ_U in proportion to n_{off} and n_{on}, but it is used on comparisons drawn from the policy-induced distribution ρ_{on}. The mismatch at a comparison z is measured by the pointwise ratio $r_U(z)$, and the coverage constant of Assumption 1 is its supremum:

$$r_U(z) = \frac{\rho_{\text{on}}(z)}{\rho^{U}_{n_{\text{off}},n_{\text{on}}}(z)}, \qquad C_{\text{cov}}(U) = \sup_{z:\,\rho^{U}_{n_{\text{off}},n_{\text{on}}}(z)>0} r_U(z).$$

Large ratios mark comparisons that the online policy produces often but that are rare in the preference training data. Random querying can spend labels on comparisons that are redundant or carry little mass under ρ_{on}, leaving such regions weakly covered. Every candidate contains a recent on-policy segment (Section 4), and among the candidates the score $U_t(z) = U^{\text{ens}}_t(z) + U^{\text{IG}}_t(z)$ favors comparisons that are ambiguous under the reward ensemble (U^{ens}_t) and informative in reward-model directions not yet covered by earlier queries (U^{IG}_t). The score can therefore move online labels toward comparisons at which the ratio would otherwise be large, and so reduce $C_{\text{cov}}(U)$ relative to random querying, as in the synthetic example of Figure 3. It need not do so: the constant decreases only if the selected comparisons have mass under ρ_{on} and are rare offline, without leaving another policy-relevant region poorly covered. Appendix D.7 gives a worked example in which the constant decreases and one in which it increases, and Appendix D.8 estimates the coverage mismatch empirically.

B.2 Detailed Comparison with Random Querying

Let ρ_U be the online query distribution of the active score, defined as in Section 5, and let ρ_{rand} be the online query distribution defined in the same way when the K pairs of each round are drawn uniformly at random, instead of by the top-K rule, from candidate pools built in the same way. With $w = n_{\text{on}}/(n_{\text{off}} + n_{\text{on}})$ and a common evaluation distribution ρ_{on}, the two rules induce the combined training distributions and coverage constants

$$\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}} = (1-w)\rho_{\text{off}} + w\rho_{\text{rand}}, \qquad \rho^{U}_{n_{\text{off}},n_{\text{on}}} = (1-w)\rho_{\text{off}} + w\rho_U,$$

$$C_{\text{cov}}(\text{rand}) = \sup_{z:\,\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}}(z)>0} \frac{\rho_{\text{on}}(z)}{\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}}(z)}, \qquad C_{\text{cov}}(U) = \sup_{z:\,\rho^{U}_{n_{\text{off}},n_{\text{on}}}(z)>0} \frac{\rho_{\text{on}}(z)}{\rho^{U}_{n_{\text{off}},n_{\text{on}}}(z)},$$

that is, the smallest constants that satisfy the inequality of Assumption 1 for the realized distributions under each rule; by the remark after the proof of Theorem 1, the theorem holds with these constants. The comparison between the two rules rests on the following assumption.

Assumption 4 (Active coverage improvement)**.** *There exists* $\alpha \in (0,1)$ *such that* $C_{\text{cov}}(U) \le \alpha C_{\text{cov}}(\text{rand})$.

Assumption 4 is a condition on the query rule, the task, and the budget. Our analysis does not prove it: ρ_U depends on the current policy and on the query history and has no closed form. The assumption can also fail. It requires the rule to place more online labels than random querying wherever ρ_{on} exceeds $\alpha C_{\text{cov}}(\text{rand})\,\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}}$, i.e., where random querying covers ρ_{on} worst, without leaving another policy-relevant region poorly covered; Appendix A.6 gives the exact condition. The score is designed for this case: every candidate contains a recent on-policy segment, so every query involves current

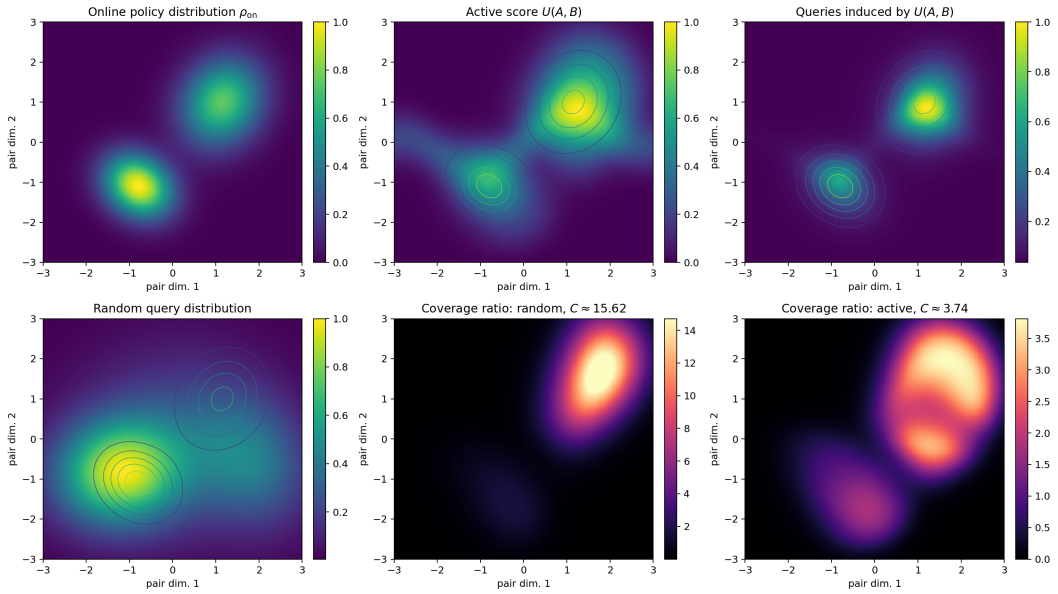

Figure 3: Schematic illustration of how a query rule shapes the online query distribution on a synthetic two-dimensional comparison space (axes: pair dim. 1 and 2). Top row: a policy-induced comparison distribution ρ_{on}, an illustrative acquisition-score surface, and the query distribution obtained by favoring high-scoring pairs. Bottom row: the query distribution of random querying, and the pointwise coverage ratio of ρ_{on} to the combined offline–online training distribution under random and under active querying (smaller is better). The constants $C \approx 15.62$ and $C \approx 3.74$ are the suprema of these ratios in this synthetic example. Whether a query rule reduces the coverage constant depends on the task (Appendices D.7 and D.8).

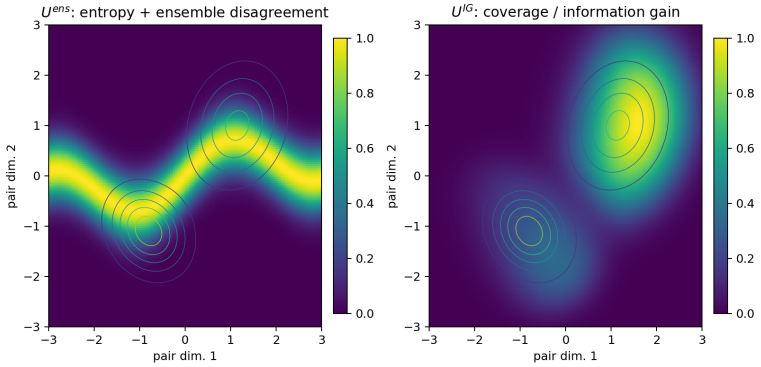

Figure 4: The two components of the query score $U = U^{\text{ens}} + U^{\text{IG}}$ on a synthetic two-dimensional comparison space (schematic). Left: the ensemble-uncertainty term U^{ens} (binary entropy of the ensemble-mean prediction plus ensemble disagreement) is high on ambiguous or disputed comparisons. Right: the information-gain term U^{IG} is high on comparisons that are informative in reward-model directions not yet covered by earlier queries.

policy behavior, and among the candidates U_t^{ens} selects ambiguous or disputed comparisons and U_t^{IG} selects comparisons that are informative in reward-model directions not covered by earlier queries. Appendix D.7 gives a two-region worked example in which the assumption holds with $\alpha = 0.6$, and a failure case in which the policy concentrates on the other region, $C_{\text{cov}}(U)/C_{\text{cov}}(\text{rand}) = 5/3 > 1$, and the assumption fails. Appendix D.8 estimates the coverage mismatch with a coarse clustered proxy. On TRIPLE-TASK4, where the query rule also changes final success, the ratio of the proxy under the active score to that under random querying falls from 0.85 at 256 online queries to 0.68 at $1{,}024$, and the error bars separate at the full budget; on TRIPLE-TASK1, where final success is

insensitive to the rule, the error bars overlap at every checkpoint. This evidence is correlational and covers two tasks.

Corollary 1 (Comparison with random querying)**.** *Suppose the conditions of Theorem 1 hold for the active rule and, with ρ_{rand} in place of ρ_U, for the random rule, and suppose Assumption 4 holds. Let $\hat{\phi}_U$ and $\hat{\phi}_{\mathrm{rand}}$ be the BT estimators trained with active and with random online queries, respectively. Then, each with probability at least $1-\delta$,*

$$\mathbb{E}_{z\sim\rho_{\mathrm{on}}}\left[|p_{\hat{\phi}_U}(z)-p^\star(z)|^2\right] \le \alpha\,\tilde{O}\left(C_{\mathrm{cov}}(\mathrm{rand})\sqrt{\frac{d_{\mathrm{eff}}+\log(1/\delta)}{n_{\mathrm{off}}+n_{\mathrm{on}}}}\right),$$

$$\mathbb{E}_{z\sim\rho_{\mathrm{on}}}\left[|p_{\hat{\phi}_{\mathrm{rand}}}(z)-p^\star(z)|^2\right] \le \tilde{O}\left(C_{\mathrm{cov}}(\mathrm{rand})\sqrt{\frac{d_{\mathrm{eff}}+\log(1/\delta)}{n_{\mathrm{off}}+n_{\mathrm{on}}}}\right).$$

Proof. Theorem 1 applied to the active rule bounds the left-hand side of the first inequality by $\tilde{O}\big(C_{\mathrm{cov}}(U)\sqrt{(d_{\mathrm{eff}}+\log(1/\delta))/(n_{\mathrm{off}}+n_{\mathrm{on}})}\big)$; substituting $C_{\mathrm{cov}}(U) \le \alpha C_{\mathrm{cov}}(\mathrm{rand})$ from Assumption 4 gives the first claim. Applying the theorem to the random rule, with $\rho^{\mathrm{rand}}_{n_{\mathrm{off}},n_{\mathrm{on}}}$ and $C_{\mathrm{cov}}(\mathrm{rand})$ in place of $\rho^{U}_{n_{\mathrm{off}},n_{\mathrm{on}}}$ and $C_{\mathrm{cov}}(U)$, gives the second. □

Under Assumption 4, the guarantee for active querying is therefore smaller than the guarantee for random querying by the factor α. The corollary compares upper bounds; it does not show that the error of random querying is larger, since its bound need not be tight.

Query-complexity consequence. Under the conditions of Corollary 1, by Proposition 2 and $C_{\mathrm{cov}}(U)^2 \le \alpha^2 C_{\mathrm{cov}}(\mathrm{rand})^2$, a target error $\mathbb{E}_{z\sim\rho_{\mathrm{on}}}[|p_{\hat{\phi}}(z)-p^\star(z)|^2] \le \epsilon^2$ is guaranteed with probability at least $1-\delta$ for active querying if

$$n^U_{\mathrm{on}} \ge \left[\tilde{O}\left(\frac{\alpha^2 C_{\mathrm{cov}}(\mathrm{rand})^2\,(d_{\mathrm{eff}}+\log(1/\delta))}{\epsilon^4}\right)-n_{\mathrm{off}}\right]_+,$$

and for random querying if

$$n^{\mathrm{rand}}_{\mathrm{on}} \ge \left[\tilde{O}\left(\frac{C_{\mathrm{cov}}(\mathrm{rand})^2\,(d_{\mathrm{eff}}+\log(1/\delta))}{\epsilon^4}\right)-n_{\mathrm{off}}\right]_+.$$

If, in addition, the coverage constants are held fixed across the budgets compared (in general, both depend on n_{off} and n_{on} through the training mixtures), the sufficient total number of offline and online labels scales by α^2: if a total $N_{\mathrm{rand}} \ge 1$ satisfies the condition $n \ge T(n)$ of (2) with $C_{\mathrm{cov}}(\mathrm{rand})$ in place of $C_{\mathrm{cov}}(U)$, then every total $n \ge \alpha^2 N_{\mathrm{rand}}$ satisfies it with $C_{\mathrm{cov}}(U)$ (see "Comparison of sufficient budgets" in Appendix A.5), so the sufficient online budget is $[\alpha^2 N_{\mathrm{rand}}-n_{\mathrm{off}}]_+$ instead of $[N_{\mathrm{rand}}-n_{\mathrm{off}}]_+$. Both are sufficient conditions derived from upper bounds, so they compare guarantees; they are not a lower bound on the number of labels that random querying needs.

B.3 Connection to Reward and Policy Convergence

The BT model identifies segment utilities only up to an additive constant, so reward differences are the identifiable object. For $z=(\sigma^A,\sigma^B)$, let $\Delta_{\hat{\phi}}(z)=\bar{R}_{\hat{\phi}}(\sigma^A)-\bar{R}_{\hat{\phi}}(\sigma^B)$ and $\Delta^\star(z)=\bar{R}^\star(\sigma^A)-\bar{R}^\star(\sigma^B)$, so that $p_{\hat{\phi}}(z)=\mathrm{sigmoid}(\Delta_{\hat{\phi}}(z))$ and $p^\star(z)=\mathrm{sigmoid}(\Delta^\star(z))$. Under Assumption 2, $|\bar{R}_\phi| \le B$ for all $\phi\in\Phi$, and realizability gives $\Delta^\star=\Delta_{\phi^\star}$ on the support of ρ_{on} (Assumption 1), so both differences lie in $[-2B,2B]$. On this interval the derivative of the sigmoid is at least $1/\kappa_B$ with $\kappa_B=2+e^{2B}+e^{-2B}$, and the mean value theorem gives $|\Delta_{\hat{\phi}}(z)-\Delta^\star(z)| \le \kappa_B|p_{\hat{\phi}}(z)-p^\star(z)|$. Combined with Theorem 1, this gives, with probability at least $1-\delta$,

$$\mathbb{E}_{z\sim\rho_{\mathrm{on}}}\left[|\Delta_{\hat{\phi}}(z)-\Delta^\star(z)|^2\right] \le \kappa_B^2\,\tilde{O}\left(C_{\mathrm{cov}}(U)\sqrt{\frac{d_{\mathrm{eff}}+\log(1/\delta)}{n_{\mathrm{off}}+n_{\mathrm{on}}}}\right).$$

This bound concerns comparisons drawn from ρ_{on}. A bound on policy suboptimality needs two further assumptions: that this reward-difference error controls the per-step reward error, up to an additive constant, on the state-action occupancies of the optimal and the learned policy (Assumption 5), and that the actor-critic procedure is approximately optimal under the learned reward (Assumption 6). Under these assumptions, Corollary 2 in Appendix D.9 gives, with probability at least $1-\delta$,

$$J^\star(\pi^\star) - J^\star(\hat{\pi}) \le \tilde{O}\left(\frac{C_{\mathrm{occ}}\,\kappa_B}{1-\gamma}\left(\frac{C_{\mathrm{cov}}(U)^2\,(d_{\mathrm{eff}} + \log(1/\delta))}{n_{\mathrm{off}} + n_{\mathrm{on}}}\right)^{1/4}\right) + \epsilon_{\mathrm{opt}}.$$

The rate is $(n_{\mathrm{off}} + n_{\mathrm{on}})^{-1/4}$ because the policy bound is linear in the reward error, which is the square root of the squared error above. If Assumption 4 also holds and Assumption 5 holds with the same C_{occ} under both rules, the first term of this upper bound is smaller than the corresponding term for random querying by the factor $\alpha^{1/2}$; without that assumption, the corollary does not compare the two rules.

B.4 Detailed Algorithm

Algorithm 1 summarizes the procedure, and Figure 1 in Section 1 gives an overview. The method first learns a reward model and policy from offline preference data, then alternates between online trajectory collection, active preference querying, reward-model updating, and policy improvement. Each round queries K pairs, so the online phase uses n_{on}/K rounds to spend the total budget n_{on}.

Algorithm 1: APOLLO — Active Preference Offline-to-onLine Learning and Optimization

Require: Offline transition dataset $\mathcal{D}_{\mathrm{off}}$, offline preference dataset $\mathcal{P}_{\mathrm{off}}$, reward ensemble $\{R_{\phi_m}\}_{m=1}^M$, critic Q_ψ, flow actor f_θ, distilled actor μ_θ, candidate-pool size N_c, queries per round K, total online query budget n_{on}.

Initialize online preference dataset $\mathcal{P}_{\mathrm{on}} \leftarrow \emptyset$.
Train reward ensemble $\{R_{\phi_m}\}_{m=1}^M$ on $\mathcal{P}_{\mathrm{off}}$ using the BT loss.
Relabel $\mathcal{D}_{\mathrm{off}}$ using the learned reward ensemble.
Pretrain critic Q_ψ, flow actor f_θ, and distilled actor μ_θ on $\mathcal{D}_{\mathrm{off}}$.
Initialize replay buffer $\mathcal{D} \leftarrow \mathcal{D}_{\mathrm{off}}$.
for online round $t = 1, \dots, n_{\mathrm{on}}/K$ **do**
  Execute the current distilled actor μ_θ to collect online trajectory segments.
  Add collected transitions to replay buffer $\mathcal{D}$.
  Construct candidate pool $\mathcal{C}_t = \{z_t^j = (\sigma_{t,j}^A, \sigma_{t,j}^B)\}_{j=1}^{N_c}$: each anchor $\sigma_{t,j}^A$ is a recent on-policy segment, and its partner $\sigma_{t,j}^B$ is sampled at random from the combined offline–online replay buffer $\mathcal{D}$.
  for each candidate pair $z \in \mathcal{C}_t$ **do**
    Compute ensemble BT predictions $\{p_{t,m}(z)\}_{m=1}^M$ and mean prediction $\bar{p}_t(z)$.
    Compute ensemble uncertainty $U_t^{\mathrm{ens}}(z)$.
    Compute information-gain score $U_t^{\mathrm{IG}}(z)$.
    Compute acquisition score $U_t(z) = U_t^{\mathrm{ens}}(z) + U_t^{\mathrm{IG}}(z)$.
  end for
  Select queried pairs $\mathcal{Q}_t = \mathrm{TopK}_{z \in \mathcal{C}_t} U_t(z)$ with $|\mathcal{Q}_t| = K$.
  Obtain preference labels $\{y_z : z \in \mathcal{Q}_t\}$ from an oracle or human annotator.
  Update online preference dataset $\mathcal{P}_{\mathrm{on}} \leftarrow \mathcal{P}_{\mathrm{on}} \cup \{(z, y_z) : z \in \mathcal{Q}_t\}$.
  Update reward ensemble on $\mathcal{P}_{\mathrm{off}} \cup \mathcal{P}_{\mathrm{on}}$.
  Relabel replay buffer $\mathcal{D}$ using the updated reward ensemble.
  Update critic Q_ψ, flow actor f_θ, and distilled actor μ_θ using the learned reward.
end for
return Reward ensemble $\{R_{\phi_m}\}_{m=1}^M$, critic Q_ψ, and distilled actor μ_θ.

B.5 Implementation Details

We implement APOLLO using an action-chunking policy parameterized by a flow-based actor. The active query module determines which online preference labels are collected, while the policy-learning module uses the learned preference reward to update the critic and actor. This design

is particularly natural for preference learning because preference labels are given over trajectory segments rather than individual transitions. Thus, predicting a short sequence of actions provides a policy representation that is well aligned with segment-level reward modeling and preference querying.

Candidate pool and query batch. At each online round, we take $N_c = 32$ recent on-policy segments as anchors and pair each with a partner segment sampled at random from the combined offline–online replay buffer, which gives N_c candidate pairs. All candidates are scored with $U_t = U_t^{\text{ens}} + U_t^{\text{IG}}$, and the top $K = 4$ are sent for labels, until the total online budget $n_{\text{on}} = 1024$ is spent (Table 4). The pairing is therefore random, but every candidate contains recent policy behavior, and the score decides which candidates are labeled.

Action chunking. Instead of outputting a single action at each environment step, the policy predicts an action chunk

$$x = (a_0, a_1, \ldots, a_{H-1}) \in \mathbb{R}^{H \times d_a},$$

where H is the action-chunk length and d_a is the action dimension. The agent executes the actions in the chunk sequentially before querying the policy again. This reduces the number of policy evaluations during online interaction and encourages temporally coherent behavior (Li et al., 2025). Action chunking is also compatible with our preference-learning setup: the reward model is trained on trajectory-segment comparisons, and an action chunk naturally corresponds to a short-horizon behavioral unit whose consequences can be evaluated by the learned preference reward.

Flow-based actor. To model potentially multi-modal action-chunk distributions, we use a flow-based actor. Flow matching provides a scalable objective for learning continuous generative models (Lipman et al., 2023), and related diffusion or flow-based policies have been effective in imitation learning and offline reinforcement learning (Chi et al., 2023; Wang et al., 2023). Let x_1 denote an action chunk sampled from the data and let $x_0 \sim \mathcal{N}(0, I)$ denote Gaussian noise of the same dimension. We define the interpolation path

$$x_u = (1-u)x_0 + ux_1, \qquad u \in [0, 1],$$

and train the flow actor $f_\theta(s, x_u, u)$ to predict the velocity $x_1 - x_0$. The flow-matching objective is

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_{s,x_1,x_0,u}\left[\|f_\theta(s, x_u, u) - (x_1 - x_0)\|_2^2\right].$$

At inference time, a multi-step flow sampler starts from Gaussian noise and integrates the learned vector field to generate an action chunk. This policy class is more expressive than a unimodal Gaussian actor and can capture multiple behavioral modes in the offline data. This is useful in preference learning, where different action chunks may be acceptable or preferred depending on the task goal and current state.

One-step distillation. Although multi-step flow sampling is expressive, it can be computationally expensive during online interaction. Following the one-step policy of Li et al. (2025), we distill the multi-step flow sampler into a one-step actor. Let $\pi_{\text{flow}}(s, \xi)$ denote the action chunk generated by the multi-step flow sampler from $\xi \sim \mathcal{N}(0, I)$. We train a distilled actor $\mu_\theta(s, \xi)$ to match the sampled action chunk by minimizing

$$\mathcal{L}_{\text{distill}}(\theta) = \mathbb{E}_{s,\xi}\left[\|\mu_\theta(s, \xi) - \pi_{\text{flow}}(s, \xi)\|_2^2\right].$$

During online execution, we use the distilled actor directly, $x = \mu_\theta(s, \xi)$, and reshape x into the action sequence $(a_0, \ldots, a_{H-1})$. In our implementation, the generated action chunk is clipped to the valid action range before execution. This distillation step preserves much of the expressiveness of the flow actor while reducing online inference cost.

Reward learning and relabeling. The reward model is learned from pairwise preference data using the BT likelihood. In the offline phase, we train the reward ensemble using $\mathcal{P}_{\text{off}}$. During online training, the active query module selects the top-K candidate pairs by score, obtains preference labels, and appends them to $\mathcal{P}_{\text{on}}$. The reward ensemble is then updated using the combined preference dataset $\mathcal{P}_{\text{off}} \cup \mathcal{P}_{\text{on}}$. After each reward-model update, we relabel the replay buffer using the updated learned reward. These inferred rewards are used for critic learning and actor improvement. Thus, preference feedback influences the policy only through the learned reward model, rather than through direct scalar environment rewards.

Critic over action chunks. Because the actor outputs action chunks, the critic is defined over state–chunk pairs, $Q_\psi(s, x)$, where $x = (a_0, \ldots, a_{H-1})$. This critic estimates the value of executing the entire action chunk from state s. The critic is first initialized from the offline dataset using rewards inferred by the offline reward model. During online training, it is further updated using the replay buffer relabeled by the current reward ensemble. This enables the value function to track changes in the learned reward model as additional online preference labels are collected.

Actor optimization. The actor objective combines behavior modeling, distillation, and value improvement. The behavior-modeling component is given by the flow-matching loss $\mathcal{L}_{\text{flow}}$. The distillation component $\mathcal{L}_{\text{distill}}$ encourages the one-step actor to imitate the multi-step flow sampler. The value-improvement term encourages the actor to generate action chunks with high critic value under the learned reward. The full actor objective is

$$\mathcal{L}_{\text{actor}}(\theta) = \mathcal{L}_{\text{flow}}(\theta) + \omega \mathcal{L}_{\text{distill}}(\theta) - \zeta \mathbb{E}_{s,\xi} \left[Q_\psi(s, \mu_\theta(s, \xi)) \right],$$

where $\omega, \zeta > 0$ are hyperparameters (Table 4). The first two terms preserve the structure and diversity of action chunks found in the offline data, while the last term shifts the actor toward action chunks that are preferred under the updated reward model.

Offline-to-online training loop. The practical training procedure consists of two phases. In the offline phase, we train the reward ensemble from $\mathcal{P}_{\text{off}}$, relabel the offline transitions with the learned reward, and pretrain the critic and actor. In the online phase, the current policy collects new trajectory segments. These segments are added to the replay buffer and serve as anchors of the candidate pairs. The active query score selects the top-K pairs for labeling. The newly labeled pairs are added to the online preference dataset, the reward ensemble is updated, the replay buffer is relabeled, and the critic and actor are further optimized.

This creates a closed feedback loop: the queried labels refine the reward model on comparisons that involve current policy behavior; the updated reward model updates the critic and actor; the updated policy generates new trajectory segments; and these segments provide new anchors for preference querying. The methodological contribution is the query rule, which decides which comparisons are labeled. The flow-based, action-chunked actor-critic with offline initialization is the policy learner through which APOLLO instantiates the rule for long-horizon control: the rule cannot create informative behavior, so the policy learner must produce trajectories whose comparisons carry useful information. Appendix D.5 replaces the flow actor, action chunking, and the policy optimizer jointly with a Gaussian Uni-IVR actor (Xu et al., 2025a) while keeping the query rule fixed. With this actor, the rule attains a higher mean success than random querying on the four tasks where the Gaussian actor has nonzero success, most clearly on TRIPLE-TASK1 (0.33 to 0.58), with differences within about one combined standard error on DOUBLE-TASK1 and DOUBLE-TASK2, but the Gaussian actor fails on DOUBLE-TASK4 and TRIPLE-TASK4 under either rule. The individual components of the actor (flow matching, action chunking, and distillation) are not ablated separately.

C Experiment setting

C.1 Environment and Dataset Information

We evaluate our method on the cube manipulation benchmarks from OGBench (Park et al., 2024), focusing on both the `cube-double` and `cube-triple` settings. In `cube-double`, the agent controls a robotic arm to manipulate two objects, whereas `cube-triple` extends the environment to three objects, increasing the combinatorial complexity of the task and making long-horizon credit assignment more challenging. For each setting, we consider four predefined single-task variants, denoted task1–task4. Each variant corresponds to a fixed goal configuration of the cubes while sharing the same transition dynamics, which provides a controlled testbed for evaluating performance across diverse manipulation objectives under consistent physical conditions. For each task, the offline preference dataset $\mathcal{P}_{\text{off}}$ contains $n_{\text{off}} = 10{,}000$ preference pairs. Each pair consists of two trajectory segments of length 50 environment steps, constructed from mixed-quality behavior data.

Simulated preference labels. Preference labels for both the offline pairs and the online queries are generated synthetically: each label indicates which of the two segments has the higher ground-truth

return. No human annotators are involved, and the agent never observes the task reward itself. Except in the label-noise study of Appendix D.11, the simulated labels are noiseless.

C.2 Baseline implementation

C.2.1 BC-ALL

BC-ALL performs behavioral cloning on all segments of the offline and online preference datasets:

$$\min_{\pi} -\mathbb{E}_{\substack{\sigma_1,\sigma_2\sim\mathcal{P}_{\text{off}}\cup\mathcal{P}_{\text{on}} \\ (s,a)\sim\sigma_1,\sigma_2}} \left[\log \pi(a|s)\right] \tag{3}$$

C.2.2 BC-PREF

BC-PREF performs behavioral cloning only on the preferred segment of each pair:

$$\min_{\pi} -\mathbb{E}_{\substack{\sigma_1,\sigma_2\sim\mathcal{P}_{\text{off}}\cup\mathcal{P}_{\text{on}},\sigma_1\succ\sigma_2 \\ (s,a)\sim\sigma_1}} \left[\log \pi(a|s)\right] \tag{4}$$

C.2.3 DPO

Direct Preference Optimization (DPO) leverages the Bradley-Terry model to optimize the policy directly from preference pairs, bypassing the need for an explicit reward model. It minimizes the negative log-likelihood of the preferred segments over the non-preferred ones:

$$\min_{\pi} -\mathbb{E}_{\substack{\sigma_1,\sigma_2\sim\mathcal{P}_{\text{off}}\cup\mathcal{P}_{\text{on}} \\ \sigma_1\succ\sigma_2}} \left[\log \text{sigmoid}\left(\beta_{\text{DPO}} \log \frac{\pi(\sigma_1)}{\pi_{\text{ref}}(\sigma_1)} - \beta_{\text{DPO}} \log \frac{\pi(\sigma_2)}{\pi_{\text{ref}}(\sigma_2)}\right)\right] \tag{5}$$

where $\beta_{\text{DPO}} > 0$ is the DPO temperature and π_{ref} is the reference policy obtained by behavioral cloning on the entire offline preference dataset. For a given trajectory segment σ, the policy likelihood is computed as the joint probability of the constituent state-action transitions.

C.2.4 IPL

We base our IPL baseline on the official implementation[1]. To adapt this inherently offline algorithm for our online fine-tuning phase, we uniformly sample trajectory pairs from the online replay buffer. These pairs are subsequently labeled and appended to the static preference dataset, continuously augmenting the training corpus during online interaction.

C.2.5 DTR

We base our DTR baseline on the official implementation[2]. Because DTR was originally formulated as a strictly offline, Transformer-based method—where a reward model is pretrained once to label the entire static dataset upfront—we introduce several modifications to adapt it for our online regime. During the online phase, as the agent interacts with the environment, we uniformly sample new trajectory pairs and append them to the active preference dataset. To accommodate this streaming data, we continuously fine-tune the reward model. Furthermore, rather than relying on a single static reward inference, we dynamically re-evaluate and infer new rewards for every training batch using the continuously updated reward model.

C.3 Hyperparameters

For our method, we adopt the core hyperparameters from Li et al. (2025). To support our active querying framework, we introduce additional hyperparameters governing the online learning phase. At each online round, the candidate pool contains N_c pairs, each formed by pairing a recent on-policy segment (the anchor) with a partner segment sampled at random from the combined offline–online replay buffer; the K candidates with the highest score U_t are queried, until the total online budget n_{on} is spent. The complete set of hyperparameters is detailed in Table 4.

[1] https://github.com/jhejna/inverse-preference-learning

[2] https://github.com/TU2021/DTR

Table 4: Hyperparameters of APOLLO used in all OGBench experiments unless stated otherwise.

Hyperparameter	**Value**
Action Chunking Length	5
Distillation Loss Weight (ω)	100 for Double-Tasks 300 for Triple-Tasks
Value-Improvement Weight (ζ)	1.0
Offline Training Steps	1,000,000
Total Online Query Budget (n_{on})	1024
Reward Ensemble Size (M)	8
Queries per Round (K)	4
Candidate Pool Size per Round (N_c)	32
Information-Gain Weight (λ_{IG})	1
Network Architecture	MLP (4 hidden layers $\times$ 512 units)
Learning Rate	3×10^{-4}
Target Soft Update (τ)	0.005
Discount Factor (γ)	0.99

C.4 Computational Resource

Our experiments were conducted using a hardware pool of 12 NVIDIA GPUs, comprising L40, A5000, and RTX 3090 models. For each experimental configuration, five training seeds were executed in parallel, sharing a single GPU allocated alongside 16 CPU cores and 64 GB of RAM. The software environment was implemented in JAX (version 0.6.0), running on CUDA 11.7 and cuDNN 8.5.0.

D Additional Experiments

This appendix provides additional empirical analyses that complement the main results. We first compare APOLLO with a purely online variant to isolate the value of offline initialization (Appendix D.1), extend the online query budget to 4096 pairs (Appendix D.2), and vary the number K of queries per round (Appendix D.3). We then study the query rule separately from the rest of the pipeline: we compare acquisition rules inside an otherwise fixed pipeline (Appendix D.4), replace the flow-based actor with a Gaussian backbone (Appendix D.5), and insert the score into purely online PbRL with PEBBLE (Appendix D.6). Appendices D.7–D.9 complement the theory with a worked example and a failure case for the coverage condition, an empirical coverage proxy, and a conditional policy-suboptimality corollary. Finally, we study the information-gain weight (Appendix D.10), noisy preference labels (Appendix D.11), the split of a fixed label budget between offline and online data (Appendix D.12), and the length of offline training (Appendix D.13).

D.1 Comparison with Online-Only Learning

To isolate the impact of the initial offline learning phase, we evaluate our full method against an online-only variant. In this setting, the agent completely bypasses the offline preference dataset ($\mathcal{P}_{\text{off}}$) and begins online environment interaction from a random network initialization, utilizing the same online query budget ($n_{\text{on}} = 1024$ pairs). The learning curves for this comparison are presented in Figure 5.

The Failure of Pure Online Preference Learning. As illustrated across all four evaluated tasks (DOUBLE-TASK1, DOUBLE-TASK4, TRIPLE-TASK1, and TRIPLE-TASK4), the online-only variant makes essentially no progress: its success rate stays near zero throughout the online phase. Because the OGBench environments consist of complex, long-horizon continuous control tasks, a randomly initialized agent is highly unlikely to stumble upon a successful—or even partially successful—sequence of actions through random exploration alone.

The Bootstrapping Problem. The failure of the online-only variant illustrates a "chicken-and-egg" problem in active preference learning: to learn a meaningful reward model, the agent must generate

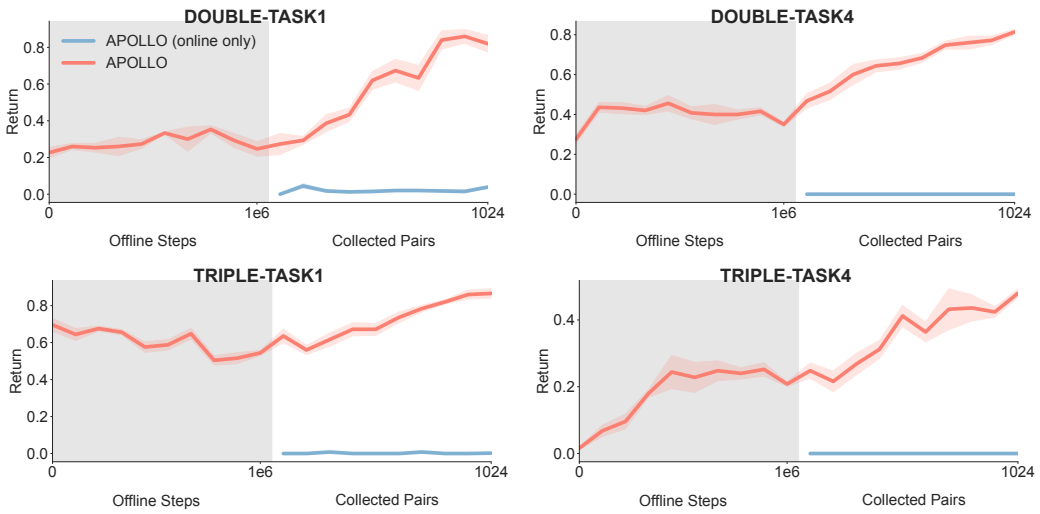

Figure 5: Performance comparison between APOLLO (red; legend "APOLLO") and its online-only variant without offline initialization (blue; legend "APOLLO (online only)") across four OGBench tasks. The shaded gray region represents the offline pre-training phase, which the online-only variant skips. Without offline initialization, online preference learning attains near-zero success within the query budget on all four tasks. The y-axis, labeled "Return" in the plots, is the task success rate in $[0, 1]$. Solid lines denote the mean over five seeds, with shaded regions indicating standard error.

informative trajectories to be queried; however, to generate informative trajectories, the agent needs a guiding reward signal. When starting from scratch, the online-only agent generates almost exclusively low-quality, out-of-distribution behavior (e.g., erratic or entirely failed movements). Comparing two equally unsuccessful trajectories yields little information about which behavior is better, so the reward model receives little signal about the task.

The Role of the Offline Prior. In contrast, APOLLO utilizes the static dataset to bootstrap both the reward ensemble and the initial flow-actor policy. When the online phase begins, the offline-initialized agent already attains nonzero success on all four tasks (Figure 5), so the on-policy anchors of the candidate pool contain partially successful behavior and the query rule can choose among informative comparisons. On these sparse, long-horizon manipulation tasks, offline initialization is therefore necessary for online preference learning to make progress within the query budget, not only a speedup.

D.2 Increasing Query Budget

To investigate the long-term stability of our active querying framework, we conduct an extended training experiment. While our main evaluations cap the online query budget at $n_{\text{on}} = 1024$ pairs to test label efficiency, here we extend the budget to 4096 collected pairs for two three-cube tasks, TRIPLE-TASK1 and TRIPLE-TASK4. The extended learning curves are presented in Figure 6.

Stability under a Larger Budget. With the larger query budget, the success rate on both tasks approaches 1.0.

- In TRIPLE-TASK1, success rises steeply once online querying begins, recovering from the decline incurred during offline training (Appendix D.13). It reaches a success rate close to 1.0 within roughly the first third of the extended online phase and stays near this level, with small spread across seeds.
- In TRIPLE-TASK4, success improves steadily throughout the extended online phase and reaches a success rate of approximately 1.0 by 4096 queries, so performance has not plateaued at the standard budget of 1024 queries.

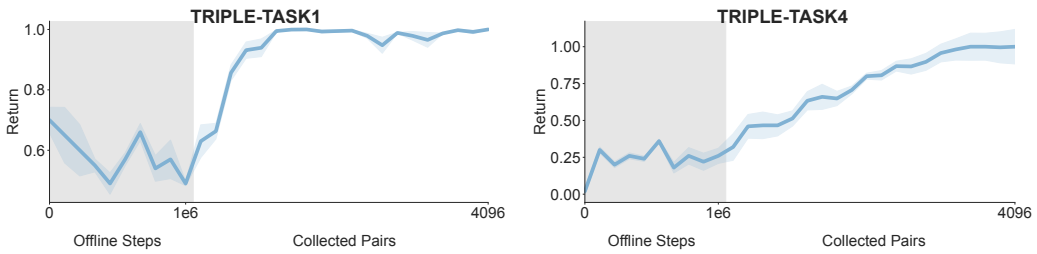

Figure 6: Extended online training performance on TRIPLE-TASK1 and TRIPLE-TASK4. The online query budget is increased from 1024 to 4096 collected pairs to evaluate long-term stability. The shaded gray region represents the initial offline pre-training phase. The y-axis, labeled "Return" in the plots, is the task success rate in $[0, 1]$. Solid lines denote the mean success rate across five seeds, with shaded regions indicating the standard error.

Neither task shows a late-stage drop in success within 4096 queries, so on these two tasks APOLLO remains stable when the budget is extended fourfold. This experiment has no random-querying counterpart, so it does not isolate the contribution of the query rule.

D.3 Ablation Study on the Top-K Parameter

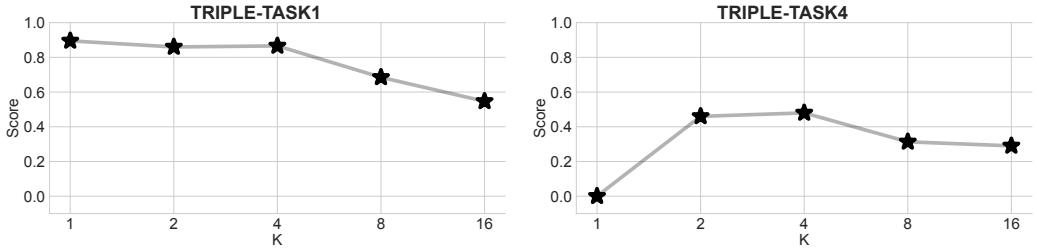

Figure 7: Ablation study on the number K of top-scoring pairs queried per round, evaluated on TRIPLE-TASK1 and TRIPLE-TASK4. The total online budget is fixed at $n_{\text{on}} = 1024$ pairs for all configurations. The y-axis, labeled "Score" in the plots, is the final success rate after online training. TRIPLE-TASK1 performs best with $K = 1$ and degrades for $K \geq 8$; TRIPLE-TASK4 fails with $K = 1$ and performs best with $K = 4$.

In our active preference learning framework, the parameter K sets how many trajectory pairs are queried in each online round. To understand the sensitivity of our method to this batch size, we conduct an ablation study varying $K \in \{1, 2, 4, 8, 16\}$. The total online budget is fixed at $n_{\text{on}} = 1024$ pairs across all runs. Therefore, altering K controls a trade-off: a smaller K gives more query rounds, each labeling only the highest-scoring candidates, while a larger K gives fewer rounds and must also label candidates with lower acquisition scores, which may increase diversity within a batch. The results on TRIPLE-TASK1 and TRIPLE-TASK4 are shown in Figure 7.

Varying Sensitivities Across Tasks. The best value of K differs between the two tasks:

- TRIPLE-TASK1: In this environment, performance is highest when K is small ($K \in \{1, 2, 4\}$), peaking at $K = 1$. Because the budget is fixed at 1024, $K = 1$ spreads it over 1024 rounds, each labeling only the top-ranked candidate. Performance decreases for $K = 8$ and $K = 16$, which is consistent with lower-ranked candidates in a larger batch being less useful for reward learning.
- TRIPLE-TASK4: This task shows a different trend. Here, $K = 1$ fails (success rate ≈ 0). A possible explanation, which we have not verified, is that updating the reward model with a

single pair per round overfits it to a narrow direction in feature space. $K = 2$ and $K = 4$ perform best, and performance drops again for $K \geq 8$.

Only the values $K \in \{2, 4\}$ are close to the best on both tasks: these values avoid the $K = 1$ failure on TRIPLE-TASK4 and the drop for $K \geq 8$ on TRIPLE-TASK1. We use $K = 4$, the best value on TRIPLE-TASK4, in all other experiments.

D.4 Comparison with Alternative Acquisition Rules

Common protocol. Unless stated otherwise, the experiments in Appendices D.4–D.13 use the OGBench tasks, offline preference datasets, and hyperparameters of Appendix C.1 and Table 4: an online budget of $n_{\text{on}} = 1024$ queried pairs, a candidate pool of $N_c = 32$ pairs per round, and $K = 4$ queries per round. Each candidate pair takes a recent on-policy segment as its anchor and pairs it with a partner segment sampled at random from the combined offline–online replay buffer; the acquisition rule then selects the top-K candidates for labeling. Every candidate therefore contains recent policy behavior, and the acquisition rule decides which of these policy-relevant comparisons are labeled. When we compare acquisition rules, the policy backbone, reward ensemble, candidate-pool construction, preference budget, and training schedule are identical, so that only the rule used to rank candidates changes. All results are averaged over five seeds and reported as mean $\pm$ standard error, except in Appendices D.12 and D.13, which report means; unless noted otherwise, they are final success rates after online training.

Acquisition rules. We compare the full score with random querying and with four alternative acquisition rules, each substituted for U_t inside the otherwise unchanged APOLLO pipeline:

- **Entropy** and **Disagreement**: the two uncertainty-based criteria of PEBBLE and B-Pref (Lee et al., 2021b;a), namely the binary entropy of the ensemble-mean prediction, $H_{\text{bin}}(\bar{p}_t(z))$, and the ensemble standard deviation, $\text{Std}_{m=1}^{M}[p_{t,m}(z)]$.
- **D-optimal**: the elliptical criterion $g_{\phi_t}(z)^\top \Lambda_t^{-1} g_{\phi_t}(z)$ that underlies the active-query RLHF method of Ji et al. (2025). It keeps only the directional part of U_t^{IG}, without the ambiguity weight $\bar{p}_t(z)(1 - \bar{p}_t(z))$ and without U_t^{ens}. Because Ji et al. (2025) target a different policy-optimization setting, this is an adaptation of their selection principle to our pipeline, not a reproduction of their full method.
- **No-IG**: the ensemble term alone, $U_t^{\text{ens}}(z)$, i.e., the sum of the Entropy and Disagreement criteria.

We run these rules inside the offline-to-online pipeline instead of running PEBBLE or B-Pref from scratch, because a from-scratch run would conflate the acquisition rule with the absence of offline initialization: on these tasks, online-only learning achieves near-zero success within the query budget (Figure 5). Appendix D.6 reports the converse experiment, in which our score is inserted into PEBBLE.

Table 5: Comparison of acquisition rules inside the APOLLO pipeline. Only the rule used to rank candidate pairs changes; the policy backbone, reward ensemble, candidate pool, query budget, and training schedule are identical. Final success rates after 1024 online queries, mean $\pm$ standard error over five seeds. Bold marks the highest mean in each row.

Task	Random	Entropy	Disagreement	D-optimal	No-IG	APOLLO
double-task1	0.01±0.00	0.29±0.26	0.31±0.29	0.37±0.29	0.40±0.19	**0.82±0.05**
double-task4	0.47±0.17	0.30±0.29	**0.81±0.03**	0.27±0.27	**0.81±0.01**	**0.81±0.01**
triple-task1	0.89±0.03	0.86±0.06	**0.91±0.03**	0.89±0.01	0.89±0.02	0.87±0.03
triple-task4	0.33±0.08	0.46±0.06	0.37±0.01	0.43±0.01	0.39±0.09	**0.48±0.01**
Mean	0.43	0.48	0.60	0.49	0.62	**0.75**
Worst task	0.01	0.29	0.31	0.27	0.39	**0.48**

Table 5 shows that APOLLO attains the highest mean success (0.75, against 0.62 for the next-best rule) and the highest worst-task success (0.48). It is not the best rule on every task. On TRIPLE-

TASK1, where random querying already reaches 0.89, all rules lie within 0.05 of one another, so this task is insensitive to the acquisition rule. The single-criterion rules are unstable on the double tasks: Entropy, Disagreement, and D-optimal all have large spread across seeds on DOUBLE-TASK1, and Entropy and D-optimal also on DOUBLE-TASK4. Summing the two uncertainty criteria (No-IG) removes the failure on DOUBLE-TASK4 but remains well below the full score on DOUBLE-TASK1 and below it on TRIPLE-TASK4. Only the combination of ensemble uncertainty with the ambiguity-weighted information-gain term is reliable across all four tasks.

D.5 Transfer to a Gaussian Policy Backbone

To test whether the benefit of the query score depends on the flow-based, action-chunked actor, we replace the actor with the single-step Gaussian backbone of Uni-RL (Uni-IVR; Xu et al., 2025a), which uses neither flow matching nor action chunking. Uni-IVR preserves replay-based offline initialization, which our setting requires, while changing both the policy family and the policy optimizer; this is why we use it instead of an on-policy method such as PPO. The reward ensemble, candidate pool, preference labels, and acquisition module are unchanged. Table 6 compares random querying with the APOLLO score under the Gaussian backbone, with the flow-based APOLLO results of Table 1 as a reference.

Table 6: Query rule under a different policy backbone. The flow-based, action-chunked actor is replaced by a Gaussian Uni-IVR actor (Xu et al., 2025a); the reward ensemble, candidate pool, labels, and acquisition module are unchanged. The last column repeats the flow-based APOLLO results of Table 1. Final success rates, mean $\pm$ standard error over five seeds. Bold marks the better Gaussian variant where they differ.

Task	Gaussian + Random	Gaussian + APOLLO	Flow APOLLO
double-task1	0.43 ± 0.05	$\mathbf{0.51\pm0.07}$	0.82 ± 0.05
double-task2	0.11 ± 0.03	$\mathbf{0.23\pm0.11}$	0.78 ± 0.02
double-task3	0.15 ± 0.07	$\mathbf{0.27\pm0.01}$	0.57 ± 0.02
double-task4	0.00 ± 0.00	0.00 ± 0.00	0.81 ± 0.01
triple-task1	0.33 ± 0.08	$\mathbf{0.58\pm0.07}$	0.87 ± 0.03
triple-task4	0.00 ± 0.00	0.00 ± 0.00	0.48 ± 0.01

Under the Gaussian backbone, the APOLLO score attains a higher mean than random querying on all four tasks where the Gaussian policy attains nonzero success. The gain is largest on TRIPLE-TASK1 ($0.33 \rightarrow 0.58$, about 2.4 combined standard errors); the gains of 0.08–0.12 on DOUBLE-TASK1–3 are smaller, and those on DOUBLE-TASK1 and DOUBLE-TASK2 are within about one combined standard error. The query rule therefore still raises mean success under a policy class without flow matching or action chunking, most clearly on TRIPLE-TASK1. (With the flow backbone, random querying instead falls to near zero within the first half of the online phase on DOUBLE-TASK1; see Figure 2.) Both Gaussian variants fail on DOUBLE-TASK4 and TRIPLE-TASK4, where the flow backbone succeeds. The query rule only decides which of the policy's comparisons are labeled, so it cannot help when the policy class does not produce successful behavior. The two components thus play complementary roles: the query score improves label efficiency for a given policy, and on the hardest tasks the flow-based, action-chunked actor is needed to generate informative behavior. Because this substitution changes flow matching, action chunking, and the policy optimizer together, it does not isolate the contribution of each actor component.

D.6 The Active Query Score in Purely Online Preference-Based RL

The previous experiments insert alternative acquisition rules into our offline-to-online pipeline. We now test the converse: whether the APOLLO score helps a standard, purely online PbRL method. Using the official B-Pref implementation of PEBBLE (Lee et al., 2021b;a), we replace only the query-selection rule and keep all other settings identical across rules. No offline preferences are used; PEBBLE's own unsupervised pretraining is the only bootstrap. We evaluate two Meta-World manipulation tasks (Yu et al., 2020) and two DeepMind Control locomotion tasks (Tassa et al., 2018), and compare with the uniform, entropy, and disagreement sampling schemes provided by B-Pref.

Table 7: The APOLLO query score inside PEBBLE (official B-Pref implementation) in a purely online setting. Only the query-selection rule changes. Meta-World tasks report success rate (%), DeepMind Control tasks report episode return; mean ± standard error over five seeds. Bold marks the highest mean in each row (Button-Press is saturated for all rules).

TASK	METRIC	UNIFORM	ENTROPY	DISAGREEMENT	APOLLO SCORE
BUTTON-PRESS-V2	SUCCESS (%)	100±0	100±0	100±0	100±0
SWEEP-INTO-V2	SUCCESS (%)	44.0±13.8	53.0±15.1	75.0±12.4	**78.0±12.1**
WALKER-WALK	RETURN	**895±13**	879±37	866±30	894±14
QUADRUPED-WALK	RETURN	307±110	399±127	510±112	**622±62**

Button-Press is saturated, and Walker-Walk does not distinguish the rules. On the two tasks where query selection matters, Sweep-Into and Quadruped-Walk, the APOLLO score attains the highest mean, although its margins over disagreement sampling on both tasks are within the error bars (78.0 ± 12.1 versus 75.0 ± 12.4 on Sweep-Into and 622 ± 62 versus 510 ± 112 on Quadruped-Walk). The score can therefore be used outside offline-to-online learning, but this experiment does not separate it from disagreement sampling. This experiment also clarifies the division of labor between the query rule and the bootstrap: the score determines which comparisons are labeled, but it does not generate informative behavior. On these benchmarks, PEBBLE's unsupervised pretraining provides enough behavioral coverage for online querying to be effective. On our sparse, long-horizon OGBench tasks, online-only learning from scratch remains at near-zero success (Figure 5), which motivates the offline bootstrap used by APOLLO.

D.7 WHEN DOES ACTIVE QUERYING REDUCE THE COVERAGE CONSTANT?

Active querying improves the bound of Theorem 1 over random querying only if it reduces the coverage constant, i.e., only if Assumption 4 holds with $\alpha < 1$. Our analysis does not establish that this holds for every task. Because queries are selected adaptively, ρ_U is an average of history-dependent query distributions: it depends on the current policy and on the query history, and it has no closed form. Here we describe the mechanism by which U_t shapes ρ_U, state the condition exactly, and give a worked example in which the coverage constant decreases and one in which it increases; Appendix D.8 estimates the coverage mismatch empirically during online training.

Mechanism. ρ_U arises in two stages. The candidate pool anchors every comparison to a recent on-policy segment, so the support of ρ_U follows the current policy. Within that support, the score assigns the top-K labels to comparisons that are ambiguous and informative in underexplored reward directions. Active querying can lower the coverage constant only if this allocation moves more label mass than random querying onto the comparisons where $\rho_{\text{on}}/\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}}$ is largest, typically comparisons that are likely under ρ_{on} but underrepresented in ρ_{off}, without leaving another policy-relevant region poorly covered (exact condition below and in Appendix A.6).

Exact condition. Let $w = n_{\text{on}}/(n_{\text{off}} + n_{\text{on}})$, so that $\rho^{U}_{n_{\text{off}},n_{\text{on}}} = (1-w)\rho_{\text{off}} + w\rho_U$. With each coverage constant taken to be the smallest value that satisfies the inequality of Assumption 1 for the realized distributions, i.e., the supremum of the realized density ratio, Appendix A.6 shows that $C_{\text{cov}}(U) \le \alpha C_{\text{cov}}(\text{rand})$ holds if and only if $w\rho_U(z) \ge \rho_{\text{on}}(z)/(\alpha C_{\text{cov}}(\text{rand})) - (1-w)\rho_{\text{off}}(z)$ for all z: the rule must add online mass wherever the offline data under-cover ρ_{on}, and ρ_U is unconstrained where $(1-w)\rho_{\text{off}}$ alone exceeds $\rho_{\text{on}}/(\alpha C_{\text{cov}}(\text{rand}))$.

Worked example. Partition the comparison space into two regions. Region 1 contains hard, policy-relevant comparisons that are rare in the offline data; Region 2 contains easier comparisons that are common offline but less relevant to the current policy. Let

$$\rho_{\text{on}} = \left(\tfrac{3}{4}, \tfrac{1}{4}\right), \qquad \rho_{\text{off}} = \left(\tfrac{1}{4}, \tfrac{3}{4}\right), \qquad \rho_{\text{rand}} = \left(\tfrac{1}{2}, \tfrac{1}{2}\right), \qquad \rho_U = (1, 0),$$

where the active rule places all online queries in Region 1 because its comparisons are both uncertain and policy-relevant. With equal offline and online label counts, $n_{\text{off}} = n_{\text{on}}$, the combined training distributions are

$$\rho^{\text{rand}}_{n_{\text{off}},n_{\text{on}}} = \left(\tfrac{3}{8}, \tfrac{5}{8}\right), \qquad \rho^{U}_{n_{\text{off}},n_{\text{on}}} = \left(\tfrac{5}{8}, \tfrac{3}{8}\right).$$

Taking the largest region-wise ratio to ρ_{on} gives

$$C_{\text{cov}}(\text{rand}) = \max\left\{\frac{3/4}{3/8}, \frac{1/4}{5/8}\right\} = 2, \qquad C_{\text{cov}}(U) = \max\left\{\frac{3/4}{5/8}, \frac{1/4}{3/8}\right\} = 1.2,$$

so Assumption 4 holds with $\alpha = 0.6$, a 40% reduction of the coverage constant, and the bound of Theorem 1 shrinks by the same factor. Coverage improves because the online labels are added where the policy has mass and the offline data are deficient.

The improvement is conditional, not universal. If the policy instead concentrates on Region 2, $\rho_{\text{on}} = (\frac{1}{4}, \frac{3}{4})$, while the rule still places its queries in Region 1, then $C_{\text{cov}}(\text{rand}) = \max\{\frac{1/4}{3/8}, \frac{3/4}{5/8}\} = 1.2$ and $C_{\text{cov}}(U) = \max\{\frac{1/4}{5/8}, \frac{3/4}{3/8}\} = 2$, so $C_{\text{cov}}(U)/C_{\text{cov}}(\text{rand}) = 5/3 > 1$: Assumption 4 fails, and the bound of Theorem 1 for active querying is $5/3$ times that for random querying (a comparison of upper bounds, not of errors). This is why we state the improvement over random querying as a consequence of Assumption 4, not as a property of the query rule that holds on every task.

Offline-heavy label budgets. The worked example assumes $n_{\text{off}} = n_{\text{on}}$, i.e., $w = 1/2$. Our experiments use $n_{\text{off}} = 10{,}000$ and $n_{\text{on}} = 1{,}024$, i.e., $w \approx 0.093$, so the online queries make up less than a tenth of the training mass. With the distributions of the first example and any $w \leq 2/3$, Region 1 attains both suprema, and $C_{\text{cov}}(\text{rand}) = 3/(1+w)$, $C_{\text{cov}}(U) = 3/(1+3w)$, and $\alpha = (1+w)/(1+3w)$. At $w \approx 0.093$, this gives $C_{\text{cov}}(\text{rand}) \approx 2.75$, $C_{\text{cov}}(U) \approx 2.35$, and $\alpha \approx 0.85$: even when every online query goes to the under-covered region, the coverage constant falls by about 15% rather than 40%, and $\alpha \to 1$ as $w \to 0$. The smaller the online share of the labels, the less the query rule can change the coverage constant.

D.8 Empirical Coverage Proxy

The coverage constant cannot be computed directly because ρ_{on} and the training distribution are continuous and unknown. We therefore estimate a discrete proxy. We embed comparison pairs with a frozen reward-model representation, partition the embedding space into 20 k-means regions $B_1, \ldots, B_{20}$, and compute

$$\widehat{C} = \max_j \frac{\hat{\rho}_{\text{on}}(B_j)}{\hat{\rho}_{\text{tr}}(B_j)},$$

where $\hat{\rho}_{\text{on}}(B_j)$ and $\hat{\rho}_{\text{tr}}(B_j)$ are the empirical fractions of policy-induced comparisons and of training comparisons (offline and queried online), respectively, that fall in B_j. The representation, the clusters, and the query counts at which $\widehat{C}$ is computed are shared by both rules; if $\hat{\rho}_{\text{on}}$ is also common to both rules, differences in $\widehat{C}$ reflect how each rule allocates training mass. Because clustering coarsens the comparison space, $\widehat{C}$ is not an estimate of C_{cov} itself, and we interpret only relative trends between the two rules.

Table 8: Empirical coverage proxy $\widehat{C}$ (lower is better) during online training, as a function of the number of online queries. Ratio rows are computed from the means. Mean ± standard error over five seeds.

		Number of online queries			
Task	Rule	256	512	768	1,024
TRIPLE-TASK1	Random	3.88±0.41	3.97±1.11	2.81±0.40	2.73±0.37
	APOLLO	3.57±0.45	3.07±0.78	2.43±0.81	2.31±0.24
	APOLLO/Random	0.92	0.77	0.86	0.85
TRIPLE-TASK4	Random	3.85±0.22	3.44±0.43	3.15±0.68	3.17±0.53
	APOLLO	3.27±0.54	2.86±0.51	2.31±0.37	2.14±0.29
	APOLLO/Random	0.85	0.83	0.73	0.68

On TRIPLE-TASK4 (Table 8), the ratio decreases monotonically from 0.85 to 0.68, and at the full budget the error bars of the two rules no longer overlap (2.14 ± 0.29 versus 3.17 ± 0.53). This

is also the task on which the query rule changes final success (0.48 versus 0.33; Table 5). On TRIPLE-TASK1 there is no such trend, the error bars overlap at every checkpoint, and final success is similar (0.87 versus 0.89). The coverage proxy therefore separates the two rules on the task where query selection affects performance and not on the task where performance is insensitive to it, which is consistent with the role of the coverage constant in Theorem 1. This evidence is correlational and covers two tasks; on TRIPLE-TASK4 it is consistent with, but does not establish, Assumption 4.

D.9 From Preference Error to Policy Suboptimality

Theorem 1 controls only the preference-prediction error $\mathbb{E}_{z\sim\rho_{\text{on}}}[|p_{\hat{\phi}}(z) - p^\star(z)|^2]$; on its own, it does not guarantee policy improvement. Converting it into a bound on policy suboptimality requires three further steps: converting preference-probability error into reward-difference error; relating segment-level reward-difference error under ρ_{on} to per-step reward error on the state-action occupancies of the relevant policies, up to the additive constant that the BT model cannot identify; and accurate policy optimization under the learned reward. The first step follows from Assumptions 1 and 2. The other two require the additional assumptions below, so the resulting corollary is conditional and is not implied by Theorem 1 alone.

Notation. For a policy π, let $d^\pi(s,a) = (1-\gamma)\sum_{t\geq 0}\gamma^t \Pr^\pi(s_t = s, a_t = a)$ be its normalized discounted occupancy, so that $J_R(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^\pi}[R(s,a)]$ for any reward R. Write $J^\star = J_{R^\star}$ for the return under the ground-truth reward, $\hat{R} = R_{\hat{\phi}}$ for the learned reward, $\pi^\star$ for an optimal policy under $R^\star$, and $\hat{\pi}$ for the policy returned by the actor-critic procedure. For $z = (\sigma^A, \sigma^B)$, let $\Delta_{\hat{\phi}}(z) = \bar{R}_{\hat{\phi}}(\sigma^A) - \bar{R}_{\hat{\phi}}(\sigma^B)$ and $\Delta^\star(z) = \bar{R}^\star(\sigma^A) - \bar{R}^\star(\sigma^B)$, so that $p_{\hat{\phi}}(z) = \text{sigmoid}(\Delta_{\hat{\phi}}(z))$ and $p^\star(z) = \text{sigmoid}(\Delta^\star(z))$.

Assumption 5 (Occupancy transfer)**.** *There exist a constant $c \in \mathbb{R}$ and a constant $C_{\text{occ}} < \infty$ such that, for $\pi \in \{\pi^\star, \hat{\pi}\}$,*

$$\mathbb{E}_{(s,a)\sim d^\pi}\left[\left|\hat{R}(s,a) - R^\star(s,a) - c\right|\right] \leq C_{\text{occ}}\left(\mathbb{E}_{z\sim\rho_{\text{on}}}\left[|\Delta_{\hat{\phi}}(z) - \Delta^\star(z)|^2\right]\right)^{1/2}.$$

Assumption 6 (Approximate policy optimization)**.** *The returned policy is ϵ_{opt}-optimal under the learned reward relative to $\pi^\star$: $J_{\hat{R}}(\pi^\star) - J_{\hat{R}}(\hat{\pi}) \leq \epsilon_{\text{opt}}$.*

Assumption 5 combines reward identifiability from pairwise comparisons with coverage of the occupancies of both $\pi^\star$ and $\hat{\pi}$ by the comparisons in ρ_{on}. This is where the unknown transition dynamics enter: our analysis conditions on the comparison distribution induced by the policy and does not provide an exploration guarantee for unknown MDPs.

Corollary 2 (Conditional policy suboptimality)**.** *Suppose the conditions of Theorem 1 and Assumptions 5–6 hold, and let $\kappa_B = 2 + e^{2B} + e^{-2B}$, where B is the utility bound of Assumption 2. Then, with probability at least $1-\delta$,*

$$J^\star(\pi^\star) - J^\star(\hat{\pi}) \leq \tilde{O}\left(\frac{C_{\text{occ}}\,\kappa_B}{1-\gamma}\left(\frac{C_{\text{cov}}(U)^2\,(d_{\text{eff}} + \log(1/\delta))}{n_{\text{off}} + n_{\text{on}}}\right)^{1/4}\right) + \epsilon_{\text{opt}}.$$

Proof. Step 1 (probabilities to reward differences). By Assumption 2, $|\bar{R}_\phi(\sigma)| \leq B$ for all $\phi \in \Phi$, so $|\Delta_{\hat{\phi}}(z)| \leq 2B$. Realizability gives $p^\star = p_{\phi^\star}$ on the support of the training distribution, and hence $\Delta^\star = \Delta_{\phi^\star}$ and $|\Delta^\star(z)| \leq 2B$ there; by Assumption 1, this support contains that of ρ_{on}. On $[-2B, 2B]$, the derivative of the sigmoid satisfies $\text{sigmoid}'(x) = \text{sigmoid}(x)(1 - \text{sigmoid}(x)) \geq 1/(2 + e^{2B} + e^{-2B}) = 1/\kappa_B$. By the mean value theorem, $|\Delta_{\hat{\phi}}(z) - \Delta^\star(z)| \leq \kappa_B |p_{\hat{\phi}}(z) - p^\star(z)|$ for ρ_{on}-almost every z, and therefore

$$\mathbb{E}_{\rho_{\text{on}}}\left[|\Delta_{\hat{\phi}}(z) - \Delta^\star(z)|^2\right] \leq \kappa_B^2\,\mathbb{E}_{\rho_{\text{on}}}\left[|p_{\hat{\phi}}(z) - p^\star(z)|^2\right].$$

Step 2 (performance difference). Since $J_{R+c}(\pi) = J_R(\pi) + c/(1-\gamma)$ for every policy, the additive constant cancels in differences of returns, and

$$J^\star(\pi^\star) - J^\star(\hat{\pi}) = \left[J_{R^\star}(\pi^\star) - J_{\hat{R}-c}(\pi^\star)\right] + \left[J_{\hat{R}}(\pi^\star) - J_{\hat{R}}(\hat{\pi})\right] + \left[J_{\hat{R}-c}(\hat{\pi}) - J_{R^\star}(\hat{\pi})\right].$$

For $\pi \in \{\pi^\star, \hat{\pi}\}$, $|J_{R^\star}(\pi) - J_{\hat{R}-c}(\pi)| \le \frac{1}{1-\gamma}\mathbb{E}_{d^\pi}|\hat{R} - R^\star - c| \le \frac{\epsilon_R}{1-\gamma}$, where ϵ_R denotes the larger of the two occupancy errors in Assumption 5. With Assumption 6,

$$J^\star(\pi^\star) - J^\star(\hat{\pi}) \le \frac{2\epsilon_R}{1-\gamma} + \epsilon_{\text{opt}}.$$

Step 3 (combination). By Assumption 5, Step 1, and Theorem 1, with probability at least $1-\delta$,

$$\epsilon_R \le C_{\text{occ}}\kappa_B \left(\mathbb{E}_{\rho_{\text{on}}}\left[|p_{\hat{\phi}}(z) - p^\star(z)|^2\right]\right)^{1/2} \le \tilde{O}\left(C_{\text{occ}}\kappa_B \left(\frac{C_{\text{cov}}(U)^2\,(d_{\text{eff}} + \log(1/\delta))}{n_{\text{off}} + n_{\text{on}}}\right)^{1/4}\right).$$

Substituting into Step 2 proves the claim. □

Because the policy bound is linear in the reward error, which is the square root of the squared preference error controlled by Theorem 1, the rate in $n_{\text{off}} + n_{\text{on}}$ is $(n_{\text{off}} + n_{\text{on}})^{-1/4}$. For a common ρ_{on} and C_{occ}, the query rule enters only through the factor $C_{\text{cov}}(U)^{1/2}$, so under Assumption 4 the first term of the upper bound is smaller than for random querying by a factor $\alpha^{1/2}$ (e.g., $0.6^{1/2} \approx 0.77$ in the worked example of Appendix D.7, and $0.85^{1/2} \approx 0.92$ for the same example at the label split of our experiments), while the ϵ_{opt} term is unchanged.

D.10 Sensitivity to the Information-Gain Weight

The default score weights its two terms equally. To test whether a different weighting helps, we evaluate

$$U_t(z; \lambda_{\text{IG}}) = U_t^{\text{ens}}(z) + \lambda_{\text{IG}}\, U_t^{\text{IG}}(z), \qquad \lambda_{\text{IG}} \in \{0, 0.5, 1, 2, 3, 5\},$$

where $\lambda_{\text{IG}} = 0$ recovers No-IG and $\lambda_{\text{IG}} = 1$ is the default used in all other experiments. The weight λ_{IG} is distinct from the regularizer λ in Λ_t.

Table 9: Sensitivity to the weight λ_{IG} of the information-gain term. $\lambda_{\text{IG}} = 0$ is No-IG and $\lambda_{\text{IG}} = 1$ is the default. Final success rates, mean ± standard error over five seeds. Bold marks the highest mean in each row.

	Information-gain weight λ_{IG}					
Task	0	0.5	1 (default)	2	3	5
triple-task1	0.89±0.02	0.89±0.06	0.87±0.03	0.89±0.04	**0.90±0.02**	0.86±0.06
triple-task4	0.39±0.09	0.43±0.12	**0.48±0.01**	0.46±0.06	0.31±0.19	0.17±0.17

TRIPLE-TASK1 is nearly insensitive to the weight (0.86–0.90 across all values; Table 9). On TRIPLE-TASK4, performance is stable for $\lambda_{\text{IG}} \in [0.5, 2]$ (0.43–0.48) and highest at the default $\lambda_{\text{IG}} = 1$. Removing the information-gain term ($\lambda_{\text{IG}} = 0$) lowers the mean to 0.39, and over-weighting it ($\lambda_{\text{IG}} \ge 3$) degrades performance to 0.31 and 0.17, with large variance across seeds. On these two tasks, equal weighting is therefore a reasonable default that did not require per-task tuning.

D.11 Robustness to Noisy Preference Labels

All other experiments use noiseless simulated preferences. To evaluate robustness to inconsistent feedback, we flip each online preference label independently with probability $\beta \in \{0.1, 0.2\}$; offline labels remain clean.

With a 20% flip rate, the first three tasks retain 94–96% of their clean-label success (Table 10). On these tasks, increasing the online query budget by about $1.3\times$ recovers the clean-label success of the standard budget (these runs are not reported in Table 10); because K is fixed, the larger budget also adds query rounds and hence optimization steps in proportion, so the recovery cannot be attributed to the extra labels alone. TRIPLE-TASK4 is the main limitation: its success drops from 0.48 ± 0.01 to 0.15 ± 0.25 at $\beta = 0.2$, several seeds collapse under noisy feedback, and these seeds do not recover within the extended budget. Symmetric random flips model annotator inconsistency, but not

Table 10: Robustness to noisy online preference labels. Each online label is flipped independently with probability β; offline labels are clean. The last column reports success at $\beta = 0.2$ as a percentage of success at $\beta = 0$. Final success rates, mean $\pm$ standard error over five seeds.

TASK	$\beta = 0$	$\beta = 0.1$	$\beta = 0.2$	RETAINED AT $\beta = 0.2$
DOUBLE-TASK1	0.82±0.05	0.79±0.12	0.78±0.09	95%
DOUBLE-TASK4	0.81±0.01	0.77±0.04	0.78±0.06	96%
TRIPLE-TASK1	0.87±0.03	0.83±0.07	0.82±0.08	94%
TRIPLE-TASK4	0.48±0.01	0.33±0.18	0.15±0.25	31%

systematic bias, temporally correlated errors, or strategic annotators. APOLLO can take human pairwise judgments through the same feedback interface; evaluation with human annotators is left to future work.

D.12 ALLOCATION BETWEEN OFFLINE AND ONLINE PREFERENCE LABELS

We vary how a fixed total of $n_{\text{off}} + n_{\text{on}} = 6{,}000$ preference labels is split between the offline dataset and online queries, using $(n_{\text{off}}, n_{\text{on}}) \in \{(5{,}000, 1{,}000), (3{,}000, 3{,}000), (1{,}000, 5{,}000)\}$.

Table 11: Allocation of a fixed total of 6,000 preference labels between offline and online data. Each entry reports success at the start of the online phase $\rightarrow$ success after online training (means over five seeds). Bold marks the highest final success in each row.

TASK	(5,000, 1,000)	(3,000, 3,000)	(1,000, 5,000)
TRIPLE-TASK1	0.49 $\rightarrow$ 0.86	0.26 $\rightarrow$ 0.91	0.23 $\rightarrow$ **1.00**
TRIPLE-TASK4	0.21 $\rightarrow$ 0.34	0.10 $\rightarrow$ **0.57**	0.01 $\rightarrow$ 0.09

TRIPLE-TASK1 benefits from the online-heavy split: despite the weakest offline initialization (0.23), allocating 5,000 labels online reaches 1.00 (Table 11), which is consistent with additional online feedback improving coverage of the policy-induced comparisons. TRIPLE-TASK4 performs best with the balanced split (0.57). With only 1,000 offline labels, its initial policy is nearly unsuccessful (0.01), likely too weak to generate informative online candidates, and the larger online budget is not exploited (0.09). There is thus no single best ratio across these tasks: the offline budget must first yield a policy capable of producing informative comparisons, after which shifting labels online can help.

D.13 EFFECT OF OFFLINE TRAINING LENGTH

The TRIPLE-TASK1 learning curves (Figure 2) drop before the online phase begins, for all query rules. The y-axis of these curves is the task success rate in $[0, 1]$ reported in Table 1. The drop occurs during offline training, before any online interaction or query: offline success peaks near 250k steps and declines to 0.54 at the default 1M steps. We hypothesize that the policy increasingly exploits errors in the learned offline reward. To test whether ending offline training earlier helps, we start the online phase from several offline checkpoints and keep everything else fixed.

Table 12: Effect of the offline training length on TRIPLE-TASK1. Online training starts from the offline checkpoint at the given step; everything else is fixed. Means over five seeds.

Offline training steps	Success at start of online phase	Final success
250k	0.71	0.98
500k	0.61	0.97
1M (default)	0.54	0.87

Ending offline training earlier improves final online success (Table 12). However, choosing the checkpoint by environment success would use the task reward, which our setting assumes is unavailable. All reported results therefore use the fixed 1M-step schedule rather than a checkpoint chosen post hoc. A reward-free alternative is to stop offline training when the BT likelihood of held-out offline preferences stops improving; we leave its evaluation to future work. Skipping the offline phase altogether still fails (Figure 5), so offline initialization remains necessary even though it can be shortened.