# Uncertainty-Guided Reward Labeling for Reinforcement Learning under Limited Feedback

**Renhao Zhang**
University of Massachusetts Amherst
renhaozhang@cs.umass.edu

**Shreyas Chaudhari**
University of Massachusetts Amherst
schaudhari@cs.umass.edu

**Bruno Castro da Silva**
University of Massachusetts Amherst
bsilva@cs.umass.edu

## Abstract

In reinforcement learning from limited feedback (RLLF), only a small fraction of an offline dataset can be labeled with rewards, and the central question is *which* samples should be labeled to learn a strong policy from the resulting partially labeled dataset. Prior work formalized this as a reward-selection problem by focusing on the selection stage while treating downstream policy learning as a black box, in a regime where queried rewards are not retained for reward-model training. We instead study the retained-label setting, where queried rewards can be stored and used to fit a reward model before policy learning. We bound the suboptimality of the learned policy by two sources of error: one from offline RL on an offline dataset, and one from reward-model uncertainty. Since reward selection cannot change the offline dataset, the limited labeling budget must be used to strategically reduce reward uncertainty. Motivated by RLLF's observation that useful rewards tend to keep the agent on high-return trajectories, we propose successor-guided uncertainty reduction (SURE), which uses successor features to select rewards that are both reachable to high-valued states and uncertainty-reducing. Theoretically, we derive SURE from a bound-induced design objective and characterize its exact one-step marginal gain. Empirically, SURE reaches near full-feedback performance with few reward labels across a variety of domains, yielding a strong method for feedback-efficient reinforcement learning.

## 1 Introduction

Modern large-scale sequential-decision problems pose a unique challenge: data can often be generated, logged, or collected at scale, but obtaining evaluative feedback for policy learning remains costly. In RLHF, models can generate many candidate responses, while feedback requires human judgment or a separate feedback oracle [1–3]; in drug discovery, generative models can propose large candidate sets, while validation requires expensive simulation or experimental evaluation [4–8]; and in offline RL, large reward-free transition datasets may be available while reward annotation remains limited [9–11]. Thus, in learning from collected data, the bottleneck often shifts from data collection to reward acquisition. When only a small fraction of the available data can be reward-labeled, the central question is *which* subset to label so that learning from the resulting partially reward-labeled dataset yields a high-performing policy.

Existing work on reinforcement learning under limited feedback (RLLF) studies reward selection in a setting where queried rewards are consumed by a policy learner rather than retained for reward-model

ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning.

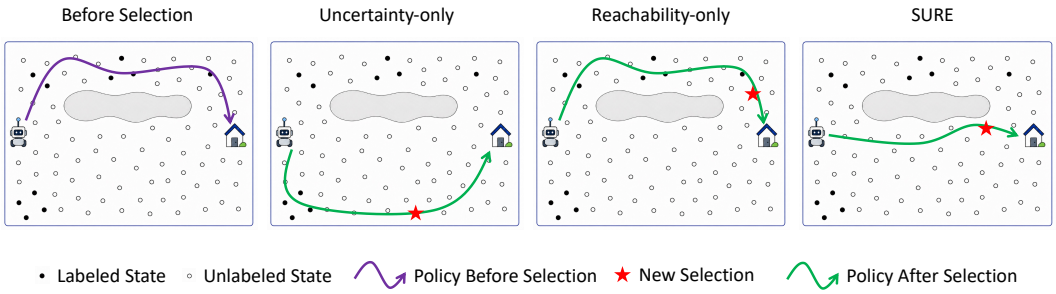


Figure 1: Illustrative retained-label reward-selection example in a navigation task. The agent must reach the goal while avoiding a central obstacle. With the initial reward labels, the policy learned from the current reward model takes a long detour, while a more direct path remains possible. Uncertainty-only selection chooses a state where the reward model is uncertain, but the selected state may be far from trajectories that lead to high return. Reachability-only selection chooses a state that can lead toward regions assigned high value by the current reward model, but this estimate can be misleading when the reward model is inaccurate. SURE combines both signals, selecting a reward that is uncertain under the current reward model and lies on a reachable path toward currently high-value states, yielding a better policy with fewer reward labels.

training [9]. This makes reward selection a question of which queried rewards directly improve policy learning. A complementary line of work on partially labeled offline RL shows that reward modeling is a useful way to leverage retained labels: rewards observed on a subset of transitions can be used to fit a reward model, which annotates the remaining offline dataset before policy optimization [10–12]. However, these methods assume that the labeled subset is passively available. They do not address which additional transitions should be selected for reward labeling. Information-directed reward learning (IDRL) is closest in that it actively queries feedback to learn a reward model for policy optimization, but the feedback is acquired online to distinguish the returns of candidate high-performing policies, rather than to decide which offline transitions should be reward-labeled [13]. This leaves open *retained-label reward selection*: actively selecting a small reward-labeled subset from a fixed offline dataset and using it to learn a reward model, which annotates the remaining data so that a high-performing policy can be learned from the resulting annotated dataset.

In this retained-label setting, selected rewards affect policy learning through the reward model they induce. A natural way to improve this model is to reduce reward uncertainty, but uncertainty reduction at different states can have very different effects on the learned policy. We formalize this intuition by showing that the performance gap between the policy learned from a partially reward-labeled dataset and the true-reward optimal policy is upper bounded by two terms: one determined by the fixed offline transition dataset, and one given by reward uncertainty under the state distribution induced by the true-reward optimal policy. Since reward selection cannot alter the fixed transition data, it can only reduce the latter term. If the optimal-policy occupancy were known, the rule would be simple: select rewards that shrink uncertainty where that occupancy places mass. Since it is unknown during selection, we introduce *Successor-Guided Uncertainty REduction* (SURE): the policy learned from the current reward model provides a moving estimate of which states are valuable, while successor features identify candidate states that can lead toward them. SURE selects rewards where reachability and uncertainty meet—states that remain uncertain and are reachable precursors to currently high-value states. Figure 1 illustrates this uncertainty–reachability tradeoff.

In this work, our contributions are:

1. Formulate *retained-label reward selection*, an offline RLLF setting where selected rewards are retained for reward-model learning and used to annotate the remaining offline data for policy learning. This extends RLLF to retained reward labels and continuous state-action domains.
2. Establish a bound-guided acquisition principle for retained-label selection. By separating the performance gap into a fixed offline-data term and a label-dependent reward-uncertainty term, our analysis identifies the ideal selection target: reducing uncertainty on states visited by the true-reward optimal policy.
3. Introduce SURE, an adaptive reward-selection rule that implements this principle by combining reward-model uncertainty with successor-guided reachability to currently high-value states.

4. Analyze and evaluate the reward-label efficiency of SURE through theory and experiments. The analysis connects selection quality to coverage mismatch and effective dimension, and experiments across navigation and locomotion domains show that SURE reaches strong performance with fewer reward labels than competing selection strategies.

## 2 Related Work

**Reward selection and active reward acquisition.** When only a limited amount of reward feedback can be obtained, different problem settings ask the learner to query different objects. Active reward learning and preference-based RL query comparisons, demonstrations, rankings, or other human feedback to identify a latent reward function [14–17]. Information-directed reward learning (IDRL) is closer to our motivation because it actively acquires feedback to learn a reward model for policy optimization [13]. However, IDRL acquires feedback during online reward learning and selects feedback that is expected to most clarify which candidate high-performing policy would achieve a higher return, whereas our setting selects which transitions in a fixed offline dataset should receive reward labels. Active RL with costly rewards and partially observable reward models also query reward observations during interaction, so acquisition is coupled to exploration and data collection [18–22]. These directions all study how to spend limited feedback, but their queried objects and learning protocols differ from retained-label offline reward selection. Closest to our setting is RLLF [9], which formulates reward selection from a reward-free offline dataset under a limited labeling budget. We build on this formulation but retain queried rewards for reward-model learning. The resulting reward-uncertainty signal guides label selection directly, avoiding the additional policy-evaluation cost of the evaluator used in RLLF.

**Partially labeled offline RL.** Partially labeled offline RL studies policy learning from offline datasets in which only a subset of transitions is annotated with rewards, while the remaining transitions are reward-free. UDS studies this setting directly and shows that simply assigning zero rewards to unlabeled transitions can be surprisingly effective, despite the bias introduced by incorrect reward labels [10]. Another strategy is to infer the missing reward signal from the labeled subset and use the learned reward to annotate reward-free transitions before applying standard offline RL [23–25]. PDS develops a pessimistic offline-RL method for partially labeled datasets by learning a conservative reward estimate from the labeled subset and using the full transition dataset for policy learning [11]. Kernelized variants extend this partially labeled offline-RL framework beyond the original linear setting [12]. Related offline policy-learning settings consider additional forms of incomplete supervision, such as observation-only data with partially labeled rewards, missing actions, low-quality demonstrations, or limited coverage [26, 27].

## 3 Problem Formulation and Preliminaries

**Preliminaries.** An MDP is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma, \eta)$, where $\mathcal{S}$ is the state space, $S_t$ is the state at time $t \in \{0, 1, \dots\}$, $\mathcal{A}$ is the action space, $A_t$ is the action at time $t$, $p(s' \mid s, a) := \Pr(S_{t+1} = s' \mid S_t = s, A_t = a)$ is the transition function, $r(s, a) \in [0, r_{\max}]$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and $\eta(s) := \Pr(S_0 = s)$ is the initial-state distribution. Specifically, we consider linear MDPs for our theoretical exposition, defined as follows.

**Definition 1** (Linear MDP [28, 29])**.** *An MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma, \eta)$ is a linear MDP if there exists a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ with $\|\phi(s, a)\|_2 \le 1$ for all $(s, a)$, an unknown vector-valued measure $\omega = (\omega_1, \dots, \omega_d)$ over $\mathcal{S}$, and an unknown reward parameter $\theta^\star \in \mathbb{R}^d$ such that $p(\cdot \mid s, a) = \phi(s, a)^\top \omega(\cdot)$ and $r(s, a) = \phi(s, a)^\top \theta^\star$ for all $(s, a)$. We assume $\|\theta^\star\|_2 \le S_r$, where $S_r > 0$ bounds the reward-parameter norm.*

For a policy $\pi(a \mid s) := \Pr(A_t = a \mid S_t = s)$, $J_r(\pi) := \mathbb{E}_\pi[\sum_{t \ge 0} \gamma^t r(s_t, a_t)]$ denotes its discounted expected return under reward function $r$. For the same MDP, with a different reward function $r'$ the discounted expected return is denoted by $J_{r'}(\pi)$. Let $\pi_r^\star \in \arg\max_\pi J_r(\pi)$ be an optimal policy under reward $r$. The discounted state–action occupancy measure of $\pi$ is $d^\pi(s, a) := (1 - \gamma) \sum_{t \ge 0} \gamma^t \Pr((s_t, a_t) = (s, a); \pi)$, which satisfies the relation $J_r(\pi) = (1 - \gamma)^{-1} \mathbb{E}_{(S,A) \sim d^\pi}[r(S, A)]$ [30].A closely related quantity is that of successor features, which encode the future state-action feature occupancy.

**Definition 2** (Successor Features [31, 32])**.** *For a policy $\pi$, the successor features of a state–action pair $(s, a)$ is defined as:*

$$\psi^{\pi}(s,a) := \mathbb{E}_{\pi}\left[\sum_{t\geq 0}\gamma^{t}\phi(S_t,A_t)\,\middle|\,(S_0,A_0)=(s,a)\right]. \tag{1}$$

Successor features represent the discounted future occupancy of the feature under a policy. They summarize which feature directions trajectories from $(s, a)$ are expected to visit.

### 3.1 Retained-Label Reward Selection

We adopt the reward-selection formulation from Chaudhari et al. [9] (RLLF). An offline dataset $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^{N}$ is obtained by the interaction of a data-collecting policy $\pi_D$ with $\mathcal{M}$, and contains no rewards. Let $\mathcal{S}_{[B]}$ denote the states that are reward-labeled: for samples in $\mathcal{D}$ where $S_t \in \mathcal{S}_{[B]}$, reward labels are assigned; the remaining samples in $\mathcal{D}$ are unlabeled. Given a labeling budget $B$, a reward-selection strategy is a map $\mathcal{Q}^{(B)} : \mathcal{D} \to \{T \subseteq \mathcal{S} : |T| = B\}$. The states selected for reward labeling are $\mathcal{S}_{[B]} = \mathcal{Q}^{(B)}(\mathcal{D})$. Unlike RLLF's original setting, where queried reward labels for $\mathcal{S}_{[B]}$ are not retained, in our setting each transition $(s_i, a_i, s'_i) \in \mathcal{D}$ with $s_i \in \mathcal{S}_{[B]}$ receives a reward label $r_i = r(s_i, a_i)$, and the resulting labeled set $\mathcal{Y}_{[B]} := \{(s_i, a_i, r_i, s'_i) : (s_i, a_i, s'_i) \in \mathcal{D},\ s_i \in \mathcal{S}_{[B]}\}$ is retained for reward-model learning, as described next.

The selected rewards are retained and used to induce a policy $\pi^{[B]}$ through the reward-modeling and offline-RL pipeline described next. The reward-selection objective is to identify a strategy $\mathcal{Q}^{(B)\star} \in \arg\max_{\mathcal{Q}^{(B)}} P(\mathcal{Q}^{(B)})$, where $P(\mathcal{Q}^{(B)}) := J_r(\pi^{[B]})$ is the return of the policy learned from the retained labels selected by $\mathcal{Q}^{(B)}$.

### 3.2 Reward Modeling for Partially Labeled Offline RL

Given a retained labeled set $\mathcal{Y}$, we follow the Provable Data Sharing framework [11]: fit a conservative reward model from $\mathcal{Y}$, use it to assign rewards to the remaining transitions in $\mathcal{D}$, and run a pessimistic offline-RL on the resulting annotated dataset. Under the linear reward model, we fit the best linear reward parameter from the labeled transitions using $\ell_2$ regularization, commonly referred to as ridge regression, $\hat{\theta}_{\mathcal{Y}} \in \arg\min_{\theta\in\mathbb{R}^d} \sum_{(s_i,a_i,r_i,s'_i)\in\mathcal{Y}} \big(\phi(s_i,a_i)^{\top}\theta - r_i\big)^2 + \lambda\|\theta\|_2^2$.

Let $\Lambda_{\mathcal{Y}} := \lambda I + \sum_{(s_i,a_i,r_i,s'_i)\in\mathcal{Y}} \phi(s_i,a_i)\phi(s_i,a_i)^{\top}$ denote the labeled reward-design matrix. The ridge solution is $\hat{\theta}_{\mathcal{Y}} = \Lambda_{\mathcal{Y}}^{-1} \sum_{(s_i,a_i,r_i,s'_i)\in\mathcal{Y}} \phi(s_i,a_i) r_i$. Using this estimate, PDS constructs a pessimistic conservative reward model

$$\tilde{r}_{\mathcal{Y}}(s,a) := \big[\phi(s,a)^{\top}\hat{\theta}_{\mathcal{Y}} - \alpha_{\mathcal{Y}}\sigma_{\mathcal{Y}}(s,a)\big]_{+},$$

where $[x]_+ := \max\{x, 0\}$ and

$$\sigma_{\mathcal{Y}}(s,a) := \sqrt{\phi(s,a)^{\top}\Lambda_{\mathcal{Y}}^{-1}\phi(s,a)}, \qquad \alpha_{\mathcal{Y}} := \sqrt{\lambda}\,S_r + \sigma_r\sqrt{2\log(1/\delta) + d\log(1+|\mathcal{Y}|/\lambda)}.$$

Here $\sigma_{\mathcal{Y}}(s,a)$ is the reward-model uncertainty induced by the retained labeled set $\mathcal{Y}$, and $\alpha_{\mathcal{Y}}$ is the corresponding confidence width, which depends on the budget $B$ and the confidence tolerance $\delta \in (0,1)$. We take $\lambda \geq 1$, so $\Lambda_{\mathcal{Y}} \succeq I$ and $\sigma_{\mathcal{Y}}(s,a) \leq \|\phi(s,a)\|_2 \leq 1$. Appendix B.1 provides the full derivation.

By the self-normalized concentration inequality of Abbasi-Yadkori et al. [33] and the nonnegative-reward clipping in Definition 1, with probability at least $1-\delta$, simultaneously over all labeled sets encountered by the learner and all $(s,a) \in \mathcal{S}\times\mathcal{A}$,

$$0 \leq r(s,a) - \tilde{r}_{\mathcal{Y}}(s,a) \leq 2\alpha_{\mathcal{Y}}\sigma_{\mathcal{Y}}(s,a). \tag{2}$$

The derivation is provided in Appendix B.1. Given the reward estimate $\tilde{r}_{\mathcal{Y}}$, the downstream learner runs a pessimistic offline-RL algorithm on $\mathcal{D}$. We use the PEVI-style backbone [34] as a black box with the following high-probability guarantee: for any pessimistic reward estimate $\tilde{r} \leq r$,

$$J_{\tilde{r}}(\pi^{\star}_{\tilde{r}}) - J_{\tilde{r}}(\hat{\pi}_{\tilde{r}}) \leq \Gamma_{\mathrm{off}}(\mathcal{D},\delta). \tag{3}$$

where $\Gamma_{\text{off}}(\mathcal{D}, \delta) := \frac{2\beta_{\text{off}}}{1-\gamma} \max_{(s,a)\in\mathcal{D}} \sqrt{\phi(s,a)^\top \Lambda(\mathcal{D})^{-1}\phi(s,a)}$, and $\beta_{\text{off}} = \widetilde{O}(d/(1-\gamma))$ is the offline confidence width. The full derivation is given in Appendix B.3. Crucially, this upper bound depends only on the fixed transition dataset $\mathcal{D}$ and confidence tolerance $\delta$, not on which transitions are reward-labeled. This motivates the reward-selection question studied next: how should the retained labels be chosen so that the resulting reward model most improves policy learning?

# 4 Successor-Guided Uncertainty Reduction

We propose *Successor-Guided Uncertainty REduction* (SURE), an adaptive reward-selection method for the retained-label RLLF setting. Following RLLF's iterative selection protocol, SURE updates the retained labeled set after each queried reward, refits the pessimistic reward estimate, and uses the resulting policy and value estimates to choose the next label. At round $b \in \{0, \dots, B-1\}$, SURE fits the current pessimistic reward $\tilde{r}_b$ from $\mathcal{Y}^{[b]}$, runs the offline-RL learner to obtain $\pi^{[b]}$, selects an unqueried transition $(s_i, a_i, s_i') \in \mathcal{D}$, queries $r_i = r(s_i, a_i)$, and updates $\mathcal{Y}^{[b+1]} = \mathcal{Y}^{[b]} \cup \{(s_i, a_i, r_i, s_i')\}$. The score used for this selection is derived below from the label-sensitive bound and then approximated with successor-guided reachability and reward-model uncertainty.

## 4.1 From the Suboptimality Bound to an Acquisition Objective

We first derive a label-sensitive upper bound on policy suboptimality. The bound separates the error due to the fixed offline transition dataset from the part that depends on the rewards selected so far.

**Theorem 1** (Label-sensitive suboptimality decomposition)**.** *Suppose the MDP is linear as in Definition 1, the retained labels produce a conservative reward estimate satisfying the reward-confidence bound (2), and the pessimistic offline-RL algorithm satisfies the stated offline-learning guarantee (3). For any $B$-budget strategy, let $\mathcal{Q}^{(b)}$ denote its first $b$ selections, let $\mathcal{Y}^{[b]}$ be the retained labeled set induced by those selections, and write $\sigma_b := \sigma_{\mathcal{Y}^{[b]}}$ and $\alpha_b := \alpha_{\mathcal{Y}^{[b]}}$. Then, with probability at least $1-2\delta$, for every round $b \in \{0, \dots, B\}$,*

$$\text{SubOpt}(\mathcal{Q}^{(b)}) \le \underbrace{\Gamma_{\text{off}}(\mathcal{D}, \delta)}_{\textit{fixed offline-data term}} + \underbrace{\frac{2\alpha_b}{1-\gamma}\sqrt{\mathbb{E}_{(s,a)\sim d^{\pi_r^\star}}[\sigma_b(s,a)^2]}}_{\textit{label-dependent reward-uncertainty term}}, \tag{4}$$

*where $\text{SubOpt}(\mathcal{Q}^{(b)}) := J_r(\pi_r^\star) - J_r(\pi^{[b]})$ is the suboptimality of $\pi^{[b]}$, the policy learned from the retained labeled set $\mathcal{Y}^{[b]}$ induced by $b$ selections.*

The proof is in Appendix C.1. Theorem 1 identifies the ideal target for uncertainty reduction as the true-reward optimal occupancy $d^{\pi_r^\star}$. Because this occupancy is unknown during selection, we need a computable surrogate, which we construct later in Section 4.2. Here, we first derive the corresponding selection objective for an arbitrary reference measure $\mu$.

For retained labeled set $\mathcal{Y}^{[b]}$ and reference measure $\mu$, define the weighted uncertainty objective

$$\Phi(\mathcal{Y}^{[b]}; \mu) := \mathbb{E}_{(s,a)\sim\mu}[\sigma_b(s,a)^2]. \tag{5}$$

This quantity measures the average squared reward uncertainty under $\mu$. In particular, Theorem 1 can be written as

$$\text{SubOpt}(\mathcal{Q}^{(b)}) \le \Gamma_{\text{off}}(\mathcal{D}, \delta) + \frac{2\alpha_b}{1-\gamma}\sqrt{\Phi(\mathcal{Y}^{[b]}; d^{\pi_r^\star})}. \tag{6}$$

With the label-dependent term isolated, we use $\Phi(\mathcal{Y}^{[b]}; \mu)$ as the objective whose reduction guides reward selection. For a fixed reference measure $\mu$, we define the **one-step acquisition function** as the uncertainty reduction obtained by reward-labeling a candidate $(s, a)$:

$$\Delta(s, a; \mathcal{Y}^{[b]}, \mu) := \Phi(\mathcal{Y}^{[b]}; \mu) - \Phi(\mathcal{Y}^{[b]} \cup \{(s,a)\}; \mu). \tag{7}$$

This marginal gain admits a closed form under the linear-MDP reward model.

**Proposition 1** (Exact one-step marginal gain)**.** *Let $\Sigma(\mu) := \mathbb{E}_{(s',a')\sim\mu}[\phi(s',a')\phi(s',a')^\top]$. For any reference measure $\mu$ and candidate $(s, a)$,*

$$\Delta(s, a; \mathcal{Y}^{[b]}, \mu) = \frac{\left\|\Sigma(\mu)^{1/2}\Lambda_b^{-1}\phi(s,a)\right\|_2^2}{1+\sigma_b(s,a)^2}. \tag{8}$$

The proof is given in Appendix C.2. Equation (8) shows that the one-step acquisition function $\Delta$ accounts for both the reference measure and the current labeled set: it favors candidates that reduce uncertainty in directions emphasized by $\mu$, while downweighting feature directions already well represented by the current labels. For fixed $\mu$, maximizing $\Delta(\cdot; \mathcal{Y}^{[b]}, \mu)$ gives the largest one-step decrease of the bound in (6). Appendix C.3 formalizes this monotonicity. We next construct a computable reference measure to replace the unavailable optimal occupancy $d^{\pi_r^\star}$.

### 4.2 Successor-Guided Surrogate Occupancy

At iteration $b$, a pessimistic reward model $r_b$ is learned from the retained labels $\mathcal{Y}^{[b]}$. The policy $\pi^{[b]}$ learned from the dataset annotated by $r_b$ provides value estimates $Q_b$ on the offline dataset. We use these estimates to construct a computable proxy for the optimal policy's occupancy: following RLLF's intuition that useful rewards should help the agent remain on, or recover toward, high-return behavior, we treat the highest-valued transitions as temporary targets, or subgoals, and use successor features to score which unlabeled candidates can reach them over multiple steps.

Let $T_b \subset \mathcal{D}$ be the top-$K$ transitions in the offline dataset ranked by $Q_b(s, a)$. For a candidate $(s_i, a_i)$ and target $(s', a')$, define the successor overlap $h_b((s_i, a_i), (s', a')) := \psi^{\pi^{[b]}}(s_i, a_i)^\top \phi(s', a')$. This overlap is large when trajectories starting from $(s_i, a_i)$ visit feature directions similar to the target under $\pi^{[b]}$. Averaging over the target set gives

$$H_b(s_i, a_i) := \frac{1}{|T_b|} \sum_{(s',a') \in T_b} h_b((s_i, a_i), (s', a')). \tag{9}$$

We normalize the positive scores over unlabeled candidates to obtain a successor-guided distribution $\bar{H}_b$; the explicit normalization is given in Appendix C.4. To avoid over-concentration when the current value estimates are inaccurate, we mix this distribution with the empirical behavior distribution $\hat{d}^{(b)}_{\text{beh}}$ over unlabeled samples:

$$\rho_b = \beta_b \hat{d}^{(b)}_{\text{beh}} + (1 - \beta_b)\bar{H}_b, \qquad \beta_b \in [0, 1]. \tag{10}$$

We refer to $\rho_b$ as the successor-guided reachability measure.

The measure $\rho_b$ is the computable reference measure used in place of the unavailable optimal occupancy $d^{\pi_r^\star}$. The behavior mixture acts as a coverage safeguard: it prevents successor guidance from concentrating entirely on the current value estimate and preserves feature directions covered by the empirical behavior distribution. Appendix D provides the corresponding theoretical analysis: reference measures constructed from the offline dataset transfer to the original optimal-occupancy objective, with guarantees controlled by coverage of the relevant feature directions.

In principle, one could plug the reachability measure $\rho_b$ into the exact gain and select $\arg\max_{(s_i,a_i) \in \mathcal{D} \setminus \mathcal{S}_b} \Delta(s_i, a_i; \mathcal{Y}^{[b]}, \rho_b)$. However, evaluating this exact gain for every candidate requires applying $\Sigma(\rho_b)^{1/2}\Lambda_b^{-1}$ to each candidate feature $\phi(s_i, a_i)$, so that the score accounts for uncertainty reduction in feature directions emphasized by $\rho_b$. Forming this matrix and scoring all unlabeled candidates has per-round computational complexity $O(d^3 + (|\mathcal{D}| - b)d^2)$ (Appendix C.5). To avoid this candidate-wise matrix-vector scoring, we define the computable SURE score as

$$\text{SURE}_b(s_i, a_i) = \rho_b(s_i, a_i) \frac{\sigma_b(s_i, a_i)^2}{1 + \sigma_b(s_i, a_i)^2}. \tag{11}$$

The first factor is the successor-guided reachability measure, which prioritizes candidates that can reach currently high-value targets while maintaining behavior-policy coverage. The second factor is the local uncertainty-reduction term, which prioritizes candidates that remain uncertain under the current reward model, with the same normalization used in Proposition 1. SURE combines these factors to select rewards that are both reachable to currently high-value targets and informative for reward-model learning, thereby targeting uncertainty reductions that are more likely to improve the learned policy. Section 5.3 shows that either factor alone can fail: REACH, which selects only by the reachability factor, can fail when current value estimates are inaccurate, while UNCERTAINTY, which selects only by the uncertainty factor, can fail when high uncertainty lies in regions unrelated to high-return behavior. Appendix A gives the full adaptive procedure.

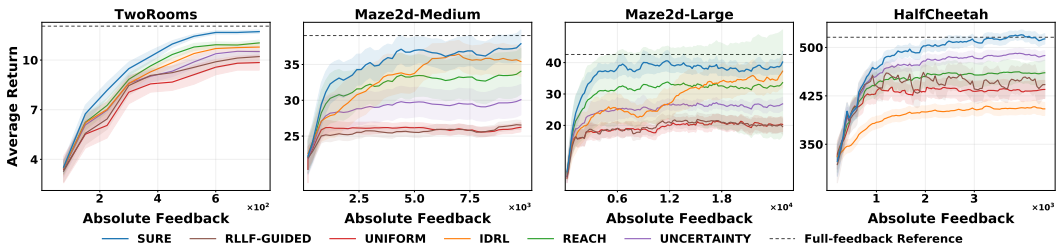


Figure 2: Performance comparison across domains under limited feedback. Return is shown as a function of the absolute feedback budget $B$. We compare SURE with UNIFORM, REACH, UNCERTAINTY, IDRL, and RLLF-GUIDED. Solid lines show the mean over 50 trials and shaded regions show the standard error. The dashed line indicates the *full-feedback reference*.

# 5 Experiments

We evaluate SURE in the retained-label RLLF setting considered in Section 3, where the learner is given a fixed reward-free offline transition dataset $\mathcal{D}$ and can acquire rewards for only $B$ selected transitions. Our experiments address three questions: (**Q1** Section 5.1) how SURE performs relative to the other reward-selection strategies across domains and budgets; (**Q2** Section 5.2) how much feedback each strategy requires to approach the full-feedback performance obtained when rewards are available for all transitions in $\mathcal{D}$; and (**Q3** Section 5.3) how each component of SURE contributes to acquisition, by ablating the successor-guided relevance factor and the uncertainty-reduction factor in equation (11).

We evaluate on four continuous state-action domains with different acquisition challenges: TwoRooms, Maze2d-Medium, and Maze2d-Large, three navigation tasks with bottleneck and goal-reaching rewards; and HalfCheetah, a high-dimensional locomotion task [35]. For each domain, all strategies use the same fixed reward-free dataset $\mathcal{D}$: we collect the TwoRooms dataset ourselves and use the D4RL datasets [36] for Maze2d-Medium, Maze2d-Large, and HalfCheetah.

Our empirical comparison focuses on reward-selection strategies: SURE selects rewards using the acquisition score in equation (11), and we compare it against alternative rules for selecting the rewards. UNIFORM samples labels uniformly from the unlabeled pool. RLLF-GUIDED adapts the guided heuristic of RLLF [9], which biases selection toward one-step predecessors of states estimated to have high value under the current learner. IDRL [13] adapts the information-directed acquisition criterion to the fixed offline dataset by estimating policy-return information with FQE. REACH uses only the successor-guided relevance factor in equation (11), while UNCERTAINTY uses only the uncertainty-reduction factor. After rewards are selected, every method follows the retained-label PDS pipeline described in Section 3: the acquired labels are used to fit a conservative reward model, which assigns rewards to the unlabeled portion of $\mathcal{D}$; IQL [37] is then run on the resulting reward-labeled dataset to produce a policy.

We report the return of learned policies as a function of the absolute feedback budget $B$ in Section 5.1 and the percentage feedback $B/|\mathcal{S}_\mathcal{D}|$ in Section 5.2, where $\mathcal{S}_\mathcal{D} := \{s_i : (s_i, a_i, s_i') \in \mathcal{D}\}$ denotes the set of distinct states in $\mathcal{D}$. We also include a *full-feedback reference*, corresponding to percentage feedback equal to one, i.e., $B = |\mathcal{S}_\mathcal{D}|$. Results are averaged over 50 trials for each strategy. Full domain and dataset details, baseline descriptions, implementation details, hyperparameters, and evaluation protocols are provided in Appendix E.

## 5.1 Label Efficiency Across Domains

Figure 2 compares SURE with the reward-selection baselines described above. The solid curve shows the mean return over 50 trials, and the shaded region shows the standard error. The dashed line indicates the full-feedback offline-RL reference, obtained by training IQL when rewards are available for all transitions in $\mathcal{D}$.

The results show that the relative performance of baselines varies across domains, while SURE remains consistently strong across all settings. Below, we highlight the main empirical findings:

Table 1: Percentage feedback needed to reach different fractions of the *full-feedback reference*. The "Fraction" column specifies the target return as a fraction of the *full-feedback reference*; "Percentage feedback required" reports the fraction of distinct states that must be labeled to reach that target. **Bold** indicates the smallest percentage feedback in each row.

| Domain | Fraction | Percentage feedback required | | | | | |
|---|---|---|---|---|---|---|---|
| | | SURE | RLLF -GUIDED | UNIFORM | IDRL | REACH | UNCERTAINTY |
| maze2d-medium | 95% | **0.20%** | 1.74% | 2.25% | 0.32% | 1.46% | 1.96% |
| | 90% | **0.12%** | 1.45% | 1.79% | 0.26% | 0.50% | 1.47% |
| | 80% | **0.05%** | 0.94% | 1.22% | 0.11% | 0.06% | 0.50% |
| maze2d-large | 95% | **0.15%** | 3.28% | 2.88% | 0.64% | 2.51% | 2.79% |
| | 90% | **0.09%** | 2.92% | 2.70% | 0.62% | 2.10% | 2.52% |
| | 80% | **0.06%** | 2.42% | 2.38% | 0.40% | 0.16% | 2.12% |
| halfcheetah | 95% | **0.71%** | 6.64% | 7.06% | 6.18% | 4.10% | 2.49% |
| | 90% | **0.45%** | 2.55% | 2.65% | 2.22% | 2.49% | 0.59% |
| | 80% | **0.30%** | 0.31% | 0.36% | 2.20% | 0.32% | 0.31% |

- **SURE is the strongest overall strategy.** Across all four domains, SURE achieves the best or near-best returns over the feedback range. The strongest competing baseline changes across domains, but SURE is the only method that remains consistently strong across the full evaluation suite.
- **Single-factor acquisition is not sufficient.** The better single-factor baseline depends on the domain. In navigation tasks, REACH is often competitive because the most useful labels are those that help identify trajectories leading toward the goal. In HalfCheetah, this spatial guidance is less informative: the higher-dimensional replay dataset makes reward uncertainty the main bottleneck, so UNCERTAINTY becomes the stronger single-factor baseline. This domain dependence shows that neither reachability nor uncertainty alone is a reliable acquisition rule.
- **Information-directed querying depends on reliable return estimates.** IDRL is competitive in the maze domains, where low-dimensional geometry and goal-reaching rewards make policy-return differences easier to estimate from limited rewards. Its performance degrades in HalfCheetah, where the higher-dimensional dataset makes the information-directed acquisition signal less reliable.
- **Multi-step reachability improves over one-step guidance.** RLLF-GUIDED often remains close to UNIFORM, suggesting that one-step predecessor guidance alone is too local for these offline datasets. In contrast, REACH is stronger in navigation domains, indicating that successor-guided multi-step reachability provides a more useful acquisition signal than selecting only immediate predecessors of currently high-value states.
- **SURE combines the useful signals without choosing one in advance.** The multiplicative acquisition score requires a candidate to be both relevant to promising behavior and informative for reward-model learning. As a result, SURE emphasizes reachability when useful trajectories are easier to identify from the geometry of the task, and emphasizes uncertainty when the relevant behaviors are harder to localize.

### 5.2 Feedback Needed to Reach Full-Feedback Performance

Table 1 converts the curves in Figure 2 into the feedback required to reach fixed fractions of the *full-feedback reference*. This view makes the cross-domain comparison more direct by normalizing for the size of each domain's state set: TwoRooms has 15K transitions, Maze2d-Medium has 2M, Maze2d-Large has 4M, and HalfCheetah has 200K.

The main takeaway is that SURE reaches the same performance levels with substantially less feedback. At the 95% level, SURE requires the least feedback in every domain: $3.17\%$ in TwoRooms, $0.20\%$ in Maze2d-Medium, $0.15\%$ in Maze2d-Large, and $0.71\%$ in HalfCheetah. Compared with the strongest non-SURE method in each domain, this corresponds to roughly $1.6\times$–$4.3\times$ less feedback. The gap is even larger against mismatched single-factor baselines: UNCERTAINTY requires about $18.6\times$ more feedback than SURE on Maze2d-Large, while REACH requires about $5.8\times$ more feedback than SURE on HalfCheetah.

Since SURE reaches the 95% level with substantially less feedback than the baselines, Figure 2 focuses on the feedback range in which SURE has largely converged. Some baselines continue to

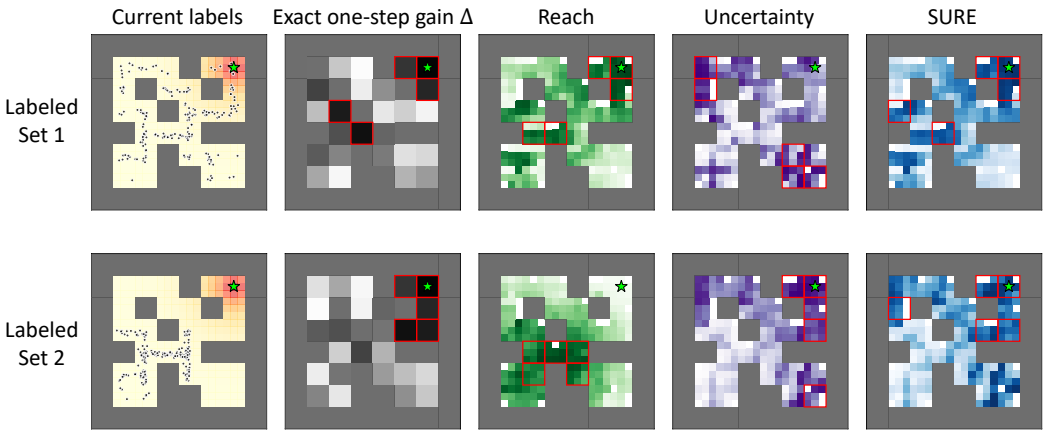


Figure 3: Ablation of the SURE score on representative maze rounds. Each row shows a different stage of selection. From left to right: acquired feedback locations, exact one-step gain $\Delta(s, a; \mathcal{Y}_{[b]}, \rho_b)$, Reach-only score, Uncertainty-only score, and SURE score. Red boxes mark the highest-scoring cells under each criterion, and the green star denotes the goal region.

improve with larger budgets, but Table 1 shows that they require much more feedback to reach the same performance levels.

### 5.3 Ablation Study of Reachability and Uncertainty

Figure 3 presents an ablation on the maze domain by visualizing how different acquisition signals rank candidate states under different initial labeled sets. Each row corresponds to one initial labeled set. The first column shows the current labeled samples (black dots) together with the underlying ground-truth reward field (background color), which is not accessible to the selection strategy. The second column shows the *exact* one-step gain, computed by adding additional feedback from each grid cell (sampling 50 transitions from the dataset whose states fall into that cell); darker colors indicate larger gain, meaning selecting from that cell under the current labeled set leads to greater policy improvement. The remaining columns show the selection distributions induced by the Reach score, the Uncertainty score, and the SURE score. Darker colors indicate higher selection density, and the highlighted cells are the top-ranked cells, i.e., the states most likely to be selected by each acquisition rule.

The results show that the alignment between each acquisition rule and the exact one-step gain depends on the structure of the current labeled set, while SURE consistently tracks the high-gain regions. When the labeled set provides broad coverage, the uncertainty field is relatively flat, and the highest-gain cells concentrate along the goal-reaching corridor; in this regime, Reach has stronger overlap with the exact gain, while Uncertainty highlights scattered regions with weaker alignment. Conversely, when the labeled set is spatially biased, the highest-gain cells shift toward previously unexplored areas; here, Uncertainty aligns better with the gain, while Reach concentrates near already-labeled regions. Across both scenarios, SURE overlaps well with the exact-gain cells by combining the two signals rather than committing to either one alone. This distinction becomes more important in higher-dimensional domains, where the labeled set cannot be easily visualized as uniformly spread or clustered, and where the labeled distribution changes as more feedback is acquired. SURE therefore provides a more robust acquisition rule by adapting to both policy relevance and reward uncertainty throughout selection.

## 6 Discussion and Conclusion

This work studies reward selection for RLLF in the retained-label setting, where queried rewards can be reused to train a reward model before downstream offline RL. Our analysis shows that, once the offline dataset is fixed, selection can only improve the reward-model uncertainty term in the suboptimality bound. This clarifies the role of reward uncertainty: uncertainty is most useful to reduce when it lies along directions that good policies are likely to use. We then propose SURE,

which turns this principle into a practical acquisition rule by combining successor-guided reachability with uncertainty reduction, so that selected rewards are both connected to promising regions and useful for shrinking the reward confidence radius. The theory justifies this rule by deriving the acquisition objective from the suboptimality bound and showing how computable reference measures can replace the unknown optimal occupancy. Empirically, SURE is consistently more label-efficient across navigation and locomotion domains, while other selection strategies are strongest mainly in regimes favored by their respective biases.

## References


[1] 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*, 2017.

[2] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[4] Jean-Louis Reymond. The chemical space project. *Accounts of Chemical Research*, 48(3): 722–730, 2015. doi: 10.1021/ar500432k.

[5] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

[6] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2323–2332, 2018.

[7] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7):eaap7885, 2018. doi: 10.1126/sciadv.aap7885.

[8] Joseph A. DiMasi, Henry G. Grabowski, and Ronald W. Hansen. Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics*, 47:20–33, 2016. doi: 10.1016/j.jhealeco.2016.01.012.

[9] Shreyas Chaudhari, Renhao Zhang, Philip S. Thomas, and Bruno Castro da Silva. Which rewards matter? reward selection for reinforcement learning under limited feedback. *arXiv preprint arXiv:2510.00144*, 2025.

[10] Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to leverage unlabeled data in offline reinforcement learning. In *International Conference on Machine Learning*, 2022.

[11] Hao Hu, Yiqin Yang, Qianchuan Zhao, and Chongjie Zhang. The provable benefits of unsupervised data sharing for offline reinforcement learning. In *International Conference on Learning Representations*, 2023.

[12] Yen-Ru Lai, Fu-Chieh Chang, and Pei-Yuan Wu. Leveraging unlabeled data sharing through kernel function approximation in offline reinforcement learning. *Transactions on Machine Learning Research*, 2025.

[13] David Lindner, Matteo Turchetta, Sebastian Tschiatschek, Kamil Ciosek, and Andreas Krause. Information directed reward learning for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

[14] Christian Daniel, Oliver Kroemer, Malte Viering, Jan Metz, and Jan Peters. Active reward learning with a novel acquisition function. *Autonomous Robots*, 39(3):389–405, 2015.

[15] Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems*, 2017.

[16] Erdem Bıyık, Malayandi Palan, Nicholas C. Landolfi, Dylan P. Losey, and Dorsa Sadigh. Asking easy questions: A user-friendly approach to active reward learning. In *Proceedings of the 3rd Conference on Robot Learning*, 2020.

[17] Nils Wilde, Dana Kulić, and Stephen L. Smith. Active preference learning using maximum regret. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[18] David Krueger, Jan Leike, Owain Evans, and John Salvatier. Active reinforcement learning: Observing rewards at a cost, 2020.

[19] Sebastian Schulze and Owain Evans. Active reinforcement learning with Monte-Carlo tree search, 2018.

[20] Aaron David Tucker, Caleb Biddulph, Claire Wang, and Thorsten Joachims. Bandits with costly reward observations. In *Conference on Uncertainty in Artificial Intelligence*, 2023.

[21] Simone Parisi, Montaser Mohammedalamen, Alireza Kazemipour, Matthew E. Taylor, and Michael Bowling. Monitored Markov decision processes. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2024.

[22] Simone Parisi, Alireza Kazemipour, and Michael Bowling. Beyond optimism: Exploration with partially observable rewards. In *Advances in Neural Information Processing Systems*, 2024.

[23] Ksenia Konyushkova, Konrad Zolna, Yusuf Aytar, Alexander Novikov, Scott Reed, Serkan Cabi, and Nando de Freitas. Semi-supervised reward learning for offline reinforcement learning, 2020.

[24] Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando de Freitas, and Scott Reed. Offline learning from demonstrations and unlabeled experience, 2020.

[25] Carlo Romeo and Andrew D. Bagdanov. Offline reinforcement learning with imputed rewards, 2024.

[26] Anqi Li, Byron Boots, and Ching-An Cheng. Mahalo: Unifying offline reinforcement learning and imitation learning from observations. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[27] Soichiro Nishimori, Xin-Qiang Cai, Johannes Ackermann, and Masashi Sugiyama. Offline reinforcement learning with domain-unlabeled data. In *Reinforcement Learning Conference*, 2025.

[28] Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10746–10756, 2020.

[29] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory (COLT)*, 2020.

[30] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[31] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

[32] André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

[33] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[34] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline reinforcement learning? In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[35] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016.

[36] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning, 2020.

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

[38] Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3703–3712, 2019.

# A SURE Algorithm

Algorithm 1 gives the full pseudocode for SURE, assembled from the SURE score (11), the reachability measure (10), the successor-guided distribution $\bar{H}_b$ of Appendix C.4 obtained by normalizing the per-candidate scores (9), and the pessimistic reward and backbone objects of Section 3. After the budget is exhausted, SURE refits the reward model using the final labeled set $\mathcal{Y}^{[B]}$, re-runs the pessimistic backbone once more on the full dataset, and returns the resulting policy $\pi^{[B]}$.

**Algorithm 1** SURE: Successor-Guided Uncertainty REduction for reward labeling

**Require:** offline dataset $\mathcal{D}$; budget $B$; top-$K$; weights $\{\beta_b\}$.

Initialize $\mathcal{Y}^{[0]} \leftarrow \emptyset$, $\mathcal{S}_0 \leftarrow \emptyset$.
**for** $b = 0, 1, \ldots, B-1$ **do**
  Fit ridge reward estimator $\hat{\theta}_b$ and build pessimistic reward $\tilde{r}_b$ from $\mathcal{Y}^{[b]}$.
  Run the offline backbone on $\mathcal{D}$ with reward $\tilde{r}_b$ to obtain $\pi^{[b]}$ and $Q_b$.
  Form target set $T_b \subset \mathcal{D}$: the top-$K$ transitions in $\mathcal{D}$ under $Q_b$.
  Compute per-candidate scores $H_b(s_i, a_i)$ via (9) and the successor-guided distribution $\bar{H}_b$ via Appendix C.4.
  Mix with behavior: $\rho_b = \beta_b \hat{d}^{(b)}_{\mathrm{beh}} + (1-\beta_b)\bar{H}_b$.
  $(s_{i_b}, a_{i_b}) \leftarrow \arg\max_{(s_i,a_i)\in\mathcal{D}\setminus\mathcal{S}_b} \mathrm{SURE}_b(s_i, a_i)$ using (11).
  Query $r_{i_b} = r(s_{i_b}, a_{i_b})$; update $\mathcal{Y}^{[b+1]} \leftarrow \mathcal{Y}^{[b]} \cup \{(s_{i_b}, a_{i_b}, r_{i_b}, s'_{i_b})\}$, $\mathcal{S}_{b+1} \leftarrow \mathcal{S}_b \cup \{(s_{i_b}, a_{i_b}, s'_{i_b})\}$.
Refit $\tilde{r}_B$ from $\mathcal{Y}^{[B]}$; run backbone on $\mathcal{D}$ with $\tilde{r}_B$ to obtain $\pi^{[B]}$.
**return** $\pi^{[B]}$.

# B Derivations for Section 3

This appendix collects the derivations of the constructions used in Section 3: the PDS pipeline that produces the pessimistic reward estimate together with the reward-confidence bound (2), a ridge estimator for the one-step feature transition operator that supports the successor features of Definition 2, and the offline-RL error term $\Gamma_{\mathrm{off}}(\mathcal{D}, \delta)$ used in equation (3).

## B.1 PDS pipeline and reward-confidence bound

The "Reward Modeling for Partially Labeled Offline RL" paragraph of Section 3 describes how the retained labeled set $\mathcal{Y}$ is converted into a reward signal that the downstream offline-RL backbone can consume. We restate that pipeline here in pragmatic terms, formalize the ridge regression step, derive the reward-confidence bound (2), and verify that the resulting $\tilde{r}_{\mathcal{Y}}$ underestimates the true reward at every state–action pair.

**The PDS pipeline.** The pipeline composes two ingredients. First, ridge regression on the labeled set $\mathcal{Y}$ (equation (3.2)) fits a single linear reward parameter $\hat{\theta}_{\mathcal{Y}}$. Because the reward model is linear in $\phi(s,a)$, this single estimator produces a prediction $\phi(s,a)^\top \hat{\theta}_{\mathcal{Y}}$ at every state–action pair—not only the queried ones—which is what allows a small number of labels to generate a reward signal everywhere the offline pool reaches. Second, PDS turns this point prediction into a lower-confidence reward

$$\tilde{r}_{\mathcal{Y}}(s,a) = \left[\phi(s,a)^\top \hat{\theta}_{\mathcal{Y}} - \alpha_{\mathcal{Y}} \sigma_{\mathcal{Y}}(s,a)\right]_+$$

by subtracting the elliptical uncertainty radius $\sigma_{\mathcal{Y}}(s,a)$ from the ridge prediction, scaled by the width $\alpha_{\mathcal{Y}}$. By construction, the result underestimates the true reward, $\tilde{r}_{\mathcal{Y}} \le r$ with high probability, so the downstream pessimistic backbone (e.g., PEVI [34]) inherits a clean reward-side guarantee.

**Matrix-form derivation of the ridge estimator.** Fix a labeled set $\mathcal{Y} \subseteq \mathcal{D}$, and let $n := |\mathcal{Y}|$. Enumerate it as $\mathcal{Y} = \{(s_1, a_1, y_1), \ldots, (s_n, a_n, y_n)\}$. Define the design matrix, the noise vector, and

the response vector

$$X := \begin{bmatrix} \phi(s_1, a_1)^\top \\ \vdots \\ \phi(s_n, a_n)^\top \end{bmatrix} \in \mathbb{R}^{n \times d}, \qquad \xi := (\xi_1, \ldots, \xi_n)^\top \in \mathbb{R}^n, \qquad y := (y_1, \ldots, y_n)^\top.$$

Under the linear reward model, $y = X\theta^\star + \xi$. In matrix form, the ridge objective (3.2) reads $L(\theta) = \|y - X\theta\|_2^2 + \lambda\|\theta\|_2^2$. Setting its gradient to zero gives the normal equation

$$\Lambda(\mathcal{Y}) = \lambda I + X^\top X, \qquad \hat{\theta}_{\mathcal{Y}} = \Lambda(\mathcal{Y})^{-1} X^\top y.$$

Substituting $y = X\theta^\star + \xi$ and using $X^\top X - \Lambda(\mathcal{Y}) = -\lambda I$, we obtain

$$\hat{\theta}_{\mathcal{Y}} - \theta^\star = \Lambda(\mathcal{Y})^{-1} X^\top \xi - \lambda \Lambda(\mathcal{Y})^{-1} \theta^\star. \tag{12}$$

The first term is the stochastic estimation error inherited from the noise $\xi$, and the second is the deterministic ridge bias toward zero induced by $\lambda$. Bounding each yields the reward-confidence inequality below.

**Derivation of the reward-confidence bound.**

*Derivation of* (2). We bound the two terms of (12) separately, combine them, and apply the nonnegative-reward clipping of PDS.

*Step 1: bounding the stochastic term.* Fix $(s, a) \in \mathcal{S} \times \mathcal{A}$. Using (12),

$$\phi(s, a)^\top (\hat{\theta}_{\mathcal{Y}} - \theta^\star) = \phi(s, a)^\top \Lambda(\mathcal{Y})^{-1} X^\top \xi - \lambda \phi(s, a)^\top \Lambda(\mathcal{Y})^{-1} \theta^\star.$$

For the first term, insert $\Lambda(\mathcal{Y})^{-1/2}\Lambda(\mathcal{Y})^{1/2}$ and apply Cauchy–Schwarz:

$$\left|\phi(s, a)^\top \Lambda(\mathcal{Y})^{-1} X^\top \xi\right| \le \|\Lambda(\mathcal{Y})^{-1/2}\phi(s, a)\|_2 \cdot \|\Lambda(\mathcal{Y})^{-1/2} X^\top \xi\|_2 = \sigma_{\mathcal{Y}}(s, a) \cdot \|X^\top \xi\|_{\Lambda(\mathcal{Y})^{-1}},$$

using $\|\Lambda(\mathcal{Y})^{-1/2}\phi(s, a)\|_2 = \sigma_{\mathcal{Y}}(s, a)$. By the self-normalized concentration inequality of Abbasi-Yadkori et al. [33] for linear regression with conditionally $\sigma_r$-sub-Gaussian noise, with probability at least $1 - \delta$, uniformly over all labeled sets $\mathcal{Y} \subseteq \mathcal{D}$ the learner may encounter,

$$\|X^\top \xi\|_{\Lambda(\mathcal{Y})^{-1}} \le \sigma_r \sqrt{2 \log\!\Big(\tfrac{\det(\Lambda(\mathcal{Y}))^{1/2}}{\det(\lambda I)^{1/2}\delta}\Big)}.$$

Bounding the determinant ratio by the AM–GM inequality together with $\sum_j \eta_j = \mathrm{tr}(X^\top X) \le n$ (since $\|\phi(s, a)\|_2 \le 1$) gives $\det(\Lambda(\mathcal{Y}))/\det(\lambda I) \le (1 + n/\lambda)^d$, hence

$$\left|\phi(s, a)^\top \Lambda(\mathcal{Y})^{-1} X^\top \xi\right| \le \sigma_{\mathcal{Y}}(s, a)\, \sigma_r \sqrt{2 \log(1/\delta) + d \log(1 + n/\lambda)}. \tag{13}$$

*Step 2: bounding the ridge-bias term.* Insert $\Lambda(\mathcal{Y})^{-1/2}\Lambda(\mathcal{Y})^{1/2}$ again and apply Cauchy–Schwarz:

$$\left|\lambda \phi(s, a)^\top \Lambda(\mathcal{Y})^{-1} \theta^\star\right| \le \sqrt{\lambda}\, \|\Lambda(\mathcal{Y})^{-1/2}\phi(s, a)\|_2 \cdot \sqrt{\lambda}\, \|\Lambda(\mathcal{Y})^{-1/2}\theta^\star\|_2 \le \sqrt{\lambda}\, S_r\, \sigma_{\mathcal{Y}}(s, a),$$

where the last step uses $\Lambda(\mathcal{Y}) \succeq \lambda I$, so $\sqrt{\lambda}\|\Lambda(\mathcal{Y})^{-1/2}\|_{\mathrm{op}} \le 1$, and $\|\theta^\star\|_2 \le S_r$.

*Step 3: combining and clipping.* Adding the two bounds,

$$\left|\phi(s, a)^\top (\hat{\theta}_{\mathcal{Y}} - \theta^\star)\right| \le \left[\sqrt{\lambda}\, S_r + \sigma_r \sqrt{2 \log(1/\delta) + d \log(1 + n/\lambda)}\right] \sigma_{\mathcal{Y}}(s, a) = \alpha_{\mathcal{Y}} \sigma_{\mathcal{Y}}(s, a).$$

Define $\hat{r}_{\mathcal{Y}}(s, a) := \phi(s, a)^\top \hat{\theta}_{\mathcal{Y}}$, so $|r(s, a) - \hat{r}_{\mathcal{Y}}(s, a)| \le \alpha_{\mathcal{Y}} \sigma_{\mathcal{Y}}(s, a)$. Recall $\tilde{r}_{\mathcal{Y}}(s, a) = \max\{\hat{r}_{\mathcal{Y}}(s, a) - \alpha_{\mathcal{Y}} \sigma_{\mathcal{Y}}(s, a), 0\}$. When the un-clipped expression is nonnegative, the displayed two-sided confidence bound gives $0 \le r(s, a) - \tilde{r}_{\mathcal{Y}}(s, a) \le 2\alpha_{\mathcal{Y}} \sigma_{\mathcal{Y}}(s, a)$; when it is negative, $\tilde{r}_{\mathcal{Y}}(s, a) = 0$ and $r(s, a) \le \hat{r}_{\mathcal{Y}}(s, a) + \alpha_{\mathcal{Y}} \sigma_{\mathcal{Y}}(s, a) < 2\alpha_{\mathcal{Y}} \sigma_{\mathcal{Y}}(s, a)$, again yielding the same two-sided gap. This is equation (2). □

### B.2 Successor-feature estimation

Under Definition 1, the successor feature of Definition 2 admits the closed form

$$\psi^{\pi}(s,a) = (I - \gamma M_{\pi})^{-1}\phi(s,a), \qquad M_{\pi} \in \mathbb{R}^{d\times d},$$

where $M_{\pi}$ is the one-step feature transition operator under $\pi$, characterized by $M_{\pi}\phi(s,a) = \mathbb{E}^{\pi}[\phi(s',a') \mid (s,a)]$ with $s' \sim p(\cdot \mid s,a)$ and $a' \sim \pi(\cdot \mid s')$. Given the offline transition dataset $\mathcal{D} = \{(s_t, a_t, s_t')\}$ and a policy $\hat{\pi}$, we estimate $M_{\hat{\pi}}$ by ridge regression of the next-feature expectation on the current feature:

$$\hat{M}_{\hat{\pi}} = \arg\min_{M\in\mathbb{R}^{d\times d}} \sum_{(s,a,s')\in\mathcal{D}} \left\|\mathbb{E}_{a'\sim\hat{\pi}(\cdot|s')}[\phi(s',a')] - M\phi(s,a)\right\|_2^2 + \lambda_{\mathrm{SF}}\|M\|_F^2, \tag{14}$$

whose closed form is the standard multivariate ridge estimator. We then set $\hat{\psi}^{\hat{\pi}}(s,a) = (I - \gamma\hat{M}_{\hat{\pi}})^{-1}\phi(s,a)$ and use $\hat{\psi}^{\hat{\pi}}$ in place of $\psi^{\hat{\pi}}$ in the per-candidate scores $H_b$ of equation (9). The estimation error $\|\hat{M}_{\hat{\pi}} - M_{\hat{\pi}}\|$ contributes an additive $\mathcal{O}(\varepsilon_{\mathrm{SF}})$ term to the unified design bound of Appendix D.1; because this term does not depend on the labeled subset $\mathcal{Y}$, it does not affect the query-complexity structure of the analysis.

### B.3 Form of the offline-RL error term

The backbone guarantee (3) is invoked abstractly in the analysis. Here we record the concrete instantiation we use in experiments—the pessimistic value-iteration (PEVI) backbone of Jin et al. [34] adapted to retained-reward partial labels by Hu et al. [11]—and derive the max-over-$\mathcal{D}$ form of $\Gamma_{\mathrm{off}}(\mathcal{D},\delta)$ used in Section 3.

Let $\Lambda(\mathcal{D}) := \lambda_{\mathrm{off}} I + \sum_{i=1}^{N}\phi(s_i,a_i)\phi(s_i,a_i)^{\top}$ be the offline value-regression design matrix built from *all* transitions in $\mathcal{D}$, with regularizer $\lambda_{\mathrm{off}} \geq 1$. The associated offline pessimism radius is $\sigma_{\mathrm{off}}(s,a) := \sqrt{\phi(s,a)^{\top}\Lambda(\mathcal{D})^{-1}\phi(s,a)}$, and the offline confidence width is

$$\beta_{\mathrm{off}} := c_{\mathrm{off}}\left(\sqrt{\lambda_{\mathrm{off}}}\,(1-\gamma)^{-1}r_{\max} + \sqrt{d\log(N/\delta)}\,(1-\gamma)^{-1}\right),$$

for an absolute constant $c_{\mathrm{off}} > 0$ inherited from the self-normalized vector concentration of Abbasi-Yadkori et al. [33]. By construction, $\beta_{\mathrm{off}}$, $\sigma_{\mathrm{off}}$, and $\Lambda(\mathcal{D})$ are functions of $(\mathcal{D},\delta)$ only.

Theorem 4.4 of Jin et al. [34], instantiated in the linear-MDP setting of Definition 1 and combined with the pessimistic-reward clipping of Hu et al. [11], yields, with probability at least $1-\delta$, the PEVI suboptimality bound

$$J_{\tilde{r}}(\pi^{\star}_{\tilde{r}}) - J_{\tilde{r}}(\hat{\pi}_{\tilde{r}}) \leq \frac{2\beta_{\mathrm{off}}}{1-\gamma}\,\mathbb{E}_{(s,a)\sim d^{\pi^{\star}_{\tilde{r}}}}\big[\sigma_{\mathrm{off}}(s,a)\big]. \tag{15}$$

The right-hand side has a residual $\tilde{r}$-dependence through the integration measure $d^{\pi^{\star}_{\tilde{r}}}$, but the integrand $\sigma_{\mathrm{off}}(s,a)$ is reward- and labeling-free: a function of $(\mathcal{D},\delta)$ only. We absorb the residual dependence by replacing the expectation with the maximum of the integrand over the transitions in $\mathcal{D}$. Pessimism in PEVI prevents the learned policy from leaving the support of $\mathcal{D}$, so $d^{\pi^{\star}_{\tilde{r}}}$ effectively concentrates on the state–action pairs $\mathcal{D}$ covers, and the chain

$$J_{\tilde{r}}(\pi^{\star}_{\tilde{r}}) - J_{\tilde{r}}(\hat{\pi}_{\tilde{r}}) \leq \frac{2\beta_{\mathrm{off}}}{1-\gamma}\,\mathbb{E}_{(s,a)\sim d^{\pi^{\star}_{\tilde{r}}}}\big[\sigma_{\mathrm{off}}(s,a)\big] \leq \frac{2\beta_{\mathrm{off}}}{1-\gamma}\max_{(s,a)\in\mathcal{D}}\sigma_{\mathrm{off}}(s,a) =: \Gamma_{\mathrm{off}}(\mathcal{D},\delta) \tag{16}$$

holds. The last equality is the definition of $\Gamma_{\mathrm{off}}(\mathcal{D},\delta)$ used in Section 3: by construction it is a function of $(\mathcal{D},\delta)$ alone, with no dependence on which transitions are reward-labeled. The full PEVI / PDS derivation of equation (15) is given by Jin et al. [34], Hu et al. [11]; we do not reproduce it here.

## C Proofs and Auxiliary Results for Section 4

This appendix proves the formal statements introduced in Section 4 and supplies the auxiliary constructions deferred from the main text. Throughout, all proofs are carried out on the intersection of the reward-confidence event of (2) and the success event of the pessimistic offline-RL backbone from (3). On this event, all displayed inequalities are deterministic.

### C.1 Proof of Theorem 1

**Theorem** (Label-sensitive suboptimality decomposition)**.** *Under Definition 1, equation (2), and equation (3), with probability at least* $1-2\delta$*, for every round* $b \in \{0,\dots,B\}$*,*

$$\mathrm{SubOpt}(\mathcal{Q}^{(b)}) \le \Gamma_{\mathrm{off}}(\mathcal{D},\delta) + \frac{2\alpha_b}{1-\gamma}\sqrt{\mathbb{E}_{(s,a)\sim d^{\pi_r^\star}}[\sigma_b(s,a)^2]}.$$

*Proof.* Write $\pi^{[b]} := \hat{\pi}_{\tilde{r}_b}$ for the policy produced by the pessimistic offline-RL backbone in round $b$. By the definition of suboptimality in Theorem 1,

$$\mathrm{SubOpt}(\mathcal{Q}^{(b)}) = J_r(\pi_r^\star) - J_r(\pi^{[b]}).$$

Add and subtract two terms involving the pessimistic reward $\tilde{r}_b$:

$$J_r(\pi_r^\star) - J_r(\pi^{[b]}) = \underbrace{\big(J_r(\pi_r^\star) - J_{\tilde{r}_b}(\pi_r^\star)\big)}_{(I)} + \underbrace{\big(J_{\tilde{r}_b}(\pi_r^\star) - J_{\tilde{r}_b}(\pi^{[b]})\big)}_{(II)} + \underbrace{\big(J_{\tilde{r}_b}(\pi^{[b]}) - J_r(\pi^{[b]})\big)}_{(III)}.$$

We bound the three terms separately.

**Step 1: bounding term** $(III)$**.** Equation (2) gives $\tilde{r}_b(s,a) \le r(s,a)$ for every $(s,a)$, and the discounted return is monotone in the reward, so $J_{\tilde{r}_b}(\pi) \le J_r(\pi)$ for any policy $\pi$. Applied to $\pi = \pi^{[b]}$,

$$(III) = J_{\tilde{r}_b}(\pi^{[b]}) - J_r(\pi^{[b]}) \le 0.$$

**Step 2: bounding term** $(II)$**.** Let $\pi^\star_{\tilde{r}_b} \in \arg\max_\pi J_{\tilde{r}_b}(\pi)$ be an optimal policy for the pessimistic reward. Since $\pi^\star_{\tilde{r}_b}$ is optimal for $\tilde{r}_b$, $J_{\tilde{r}_b}(\pi_r^\star) \le J_{\tilde{r}_b}(\pi^\star_{\tilde{r}_b})$, and therefore

$$(II) \le J_{\tilde{r}_b}(\pi^\star_{\tilde{r}_b}) - J_{\tilde{r}_b}(\pi^{[b]}).$$

With $\pi^{[b]} = \hat{\pi}_{\tilde{r}_b}$, the backbone guarantee in equation (3) gives

$$J_{\tilde{r}_b}(\pi^\star_{\tilde{r}_b}) - J_{\tilde{r}_b}(\pi^{[b]}) \le \Gamma_{\mathrm{off}}(\mathcal{D},\delta),$$

and hence $(II) \le \Gamma_{\mathrm{off}}(\mathcal{D},\delta)$.

**Step 3: bounding term** $(I)$**.** By definition of value,

$$(I) = \mathbb{E}_\eta^{\pi_r^\star}\left[\sum_{t=0}^{\infty}\gamma^t\big(r(s_t,a_t) - \tilde{r}_b(s_t,a_t)\big)\right].$$

Equation (2) gives $0 \le r(s_t,a_t) - \tilde{r}_b(s_t,a_t) \le 2\alpha_b\sigma_b(s_t,a_t)$ for every $t$, so

$$(I) \le 2\alpha_b\,\mathbb{E}_\eta^{\pi_r^\star}\left[\sum_{t=0}^{\infty}\gamma^t\sigma_b(s_t,a_t)\right].$$

The discounted occupancy identity $\mathbb{E}_\eta^{\pi_r^\star}\big[\sum_t \gamma^t f(s_t,a_t)\big] = (1-\gamma)^{-1}\mathbb{E}_{(s,a)\sim d^{\pi_r^\star}}[f(s,a)]$, applied with $f(s,a) = \sigma_b(s,a)$, yields

$$(I) \le \frac{2\alpha_b}{1-\gamma}\,\mathbb{E}_{(s,a)\sim d^{\pi_r^\star}}[\sigma_b(s,a)].$$

**Step 4: assembling the pieces.** Combining the three bounds gives

$$\mathrm{SubOpt}(\mathcal{Q}^{(b)}) \le \Gamma_{\mathrm{off}}(\mathcal{D},\delta) + \frac{2\alpha_b}{1-\gamma}\,\mathbb{E}_{(s,a)\sim d^{\pi_r^\star}}[\sigma_b(s,a)].$$

Since $\sigma_b(s,a) \ge 0$ and $d^{\pi_r^\star}$ is a probability measure, Jensen's inequality applied to the concave square-root yields $\mathbb{E}_{(s,a)\sim d^{\pi_r^\star}}[\sigma_b(s,a)] \le \sqrt{\mathbb{E}_{(s,a)\sim d^{\pi_r^\star}}[\sigma_b(s,a)^2]}$, and substitution gives the claimed bound. □

### C.2 Proof of Proposition 1

**Proposition** (Exact one-step marginal gain)**.** *For any reference measure $\mu$ over the pool and any candidate $(s, a)$,*

$$\Delta(s, a; \mathcal{Y}^{[b]}, \mu) = \Phi(\mathcal{Y}^{[b]}; \mu) - \Phi(\mathcal{Y}^{[b]} \cup \{(s, a)\}; \mu) = \frac{\|\Sigma(\mu)^{1/2}\Lambda_b^{-1}\phi(s, a)\|_2^2}{1 + \sigma_b(s, a)^2}.$$

*In particular, $\Phi(\mathcal{Y}^{[b]}; \mu)$ is monotonically nonincreasing in the labeled set.*

*Proof.* For readability, write $\Lambda := \Lambda_b$, $\Sigma := \Sigma(\mu)$, $\phi := \phi(s, a)$, and $\Lambda^+ := \Lambda + \phi\phi^\top$. The design potential admits a trace form: by definition $\sigma_b(s, a)^2 = \phi(s, a)^\top\Lambda^{-1}\phi(s, a) = \mathrm{tr}(\Lambda^{-1}\phi(s, a)\phi(s, a)^\top)$, and taking expectation over $(s, a) \sim \mu$ with linearity of trace gives

$$\Phi(\mathcal{Y}^{[b]}; \mu) = \mathbb{E}_{(s,a)\sim\mu}[\sigma_b(s, a)^2] = \mathrm{tr}\big(\Sigma(\mu)\Lambda_b^{-1}\big). \tag{17}$$

Applying (17) to both $\mathcal{Y}^{[b]}$ and $\mathcal{Y}^{[b]} \cup \{(s, a)\}$,

$$\Phi(\mathcal{Y}^{[b]}; \mu) = \mathrm{tr}(\Sigma\Lambda^{-1}), \qquad \Phi(\mathcal{Y}^{[b]} \cup \{(s, a)\}; \mu) = \mathrm{tr}(\Sigma(\Lambda^+)^{-1}),$$

so

$$\Delta(s, a; \mathcal{Y}^{[b]}, \mu) = \mathrm{tr}(\Sigma\Lambda^{-1}) - \mathrm{tr}(\Sigma(\Lambda^+)^{-1}). \tag{18}$$

**Step 1: Sherman–Morrison update.** Since $\Lambda$ is positive definite, the rank-one inverse formula applies:

$$(\Lambda + \phi\phi^\top)^{-1} = \Lambda^{-1} - \frac{\Lambda^{-1}\phi\phi^\top\Lambda^{-1}}{1 + \phi^\top\Lambda^{-1}\phi}.$$

Substituting into (18) and using linearity of trace, the $\mathrm{tr}(\Sigma\Lambda^{-1})$ terms cancel, leaving

$$\Delta(s, a; \mathcal{Y}^{[b]}, \mu) = \frac{\mathrm{tr}\big(\Sigma\Lambda^{-1}\phi\phi^\top\Lambda^{-1}\big)}{1 + \phi^\top\Lambda^{-1}\phi}.$$

**Step 2: simplifying the denominator.** By definition of the uncertainty radius, $\phi^\top\Lambda^{-1}\phi = \sigma_b(s, a)^2$, so the denominator is $1 + \sigma_b(s, a)^2$.

**Step 3: simplifying the numerator.** By cyclicity of trace and the existence of a symmetric positive semidefinite square root $\Sigma^{1/2}$,

$$\mathrm{tr}\big(\Sigma\Lambda^{-1}\phi\phi^\top\Lambda^{-1}\big) = \phi^\top\Lambda^{-1}\Sigma\Lambda^{-1}\phi = \|\Sigma^{1/2}\Lambda^{-1}\phi\|_2^2.$$

Substituting yields

$$\Delta(s, a; \mathcal{Y}^{[b]}, \mu) = \frac{\|\Sigma(\mu)^{1/2}\Lambda_b^{-1}\phi(s, a)\|_2^2}{1 + \sigma_b(s, a)^2},$$

the claimed formula.

**Step 4: monotonicity.** The numerator is a squared Euclidean norm and therefore nonnegative, and $1 + \sigma_b(s, a)^2 \geq 1 > 0$, so $\Delta(s, a; \mathcal{Y}^{[b]}, \mu) \geq 0$ for every $(s, a)$. By definition of $\Delta$, $\Phi(\mathcal{Y}^{[b]} \cup \{(s, a)\}; \mu) = \Phi(\mathcal{Y}^{[b]}; \mu) - \Delta(s, a; \mathcal{Y}^{[b]}, \mu) \leq \Phi(\mathcal{Y}^{[b]}; \mu)$, so $\Phi$ is monotonically nonincreasing in the labeled set. □

### C.3 Marginal gain and bound decrease

The bound in equation (6) is monotone in $\Phi(\mathcal{Y}^{[b]}; \mu)$ for any fixed reference measure $\mu$, so the largest one-step decrease of the bound is achieved by selecting the candidate that maximizes the marginal gain $\Delta(\cdot; \mathcal{Y}^{[b]}, \mu)$. We make this precise.

For a fixed reference measure $\mu$, define the bound-side objective

$$U_b(\mu) := \Gamma_{\text{off}}(\mathcal{D}, \delta) + \frac{2\alpha_b}{1 - \gamma}\sqrt{\Phi(\mathcal{Y}^{[b]}; \mu)}, \tag{19}$$

and let $U_b^{(s,a)}(\mu)$ denote the same quantity after adding $(s, a)$ to the labeled set.

**Lemma 1** (Marginal gain orders bound improvement)**.** *Fix a reference measure $\mu$ and round $b$. For any candidate $(s,a)$ with $\Delta(s,a;\mathcal{Y}^{[b]},\mu) \le \Phi(\mathcal{Y}^{[b]};\mu)$,*

$$U_b(\mu) - U_b^{(s,a)}(\mu) = \frac{2\alpha_b}{1-\gamma}\Big(\sqrt{\Phi(\mathcal{Y}^{[b]};\mu)} - \sqrt{\Phi(\mathcal{Y}^{[b]};\mu) - \Delta(s,a;\mathcal{Y}^{[b]},\mu)}\Big). \tag{20}$$

*Consequently, $U_b(\mu) - U_b^{(s,a)}(\mu)$ is strictly increasing in $\Delta(s,a;\mathcal{Y}^{[b]},\mu)$.*

*Proof.* By the definition of $\Delta$ in equation (7), $\Phi(\mathcal{Y}^{[b]} \cup \{(s,a)\};\mu) = \Phi(\mathcal{Y}^{[b]};\mu) - \Delta(s,a;\mathcal{Y}^{[b]},\mu)$. Substituting into (19) for $U_b$ and into the analogous expression for $U_b^{(s,a)}$ yields (20). Applying the elementary identity $a - b = (a^2 - b^2)/(a+b)$ with $a = \sqrt{\Phi(\mathcal{Y}^{[b]};\mu)}$ and $b = \sqrt{\Phi(\mathcal{Y}^{[b]};\mu) - \Delta(s,a;\mathcal{Y}^{[b]},\mu)}$ yields the equivalent ratio form

$$U_b(\mu) - U_b^{(s,a)}(\mu) = \frac{2\alpha_b}{1-\gamma} \frac{\Delta(s,a;\mathcal{Y}^{[b]},\mu)}{\sqrt{\Phi(\mathcal{Y}^{[b]};\mu)} + \sqrt{\Phi(\mathcal{Y}^{[b]};\mu) - \Delta(s,a;\mathcal{Y}^{[b]},\mu)}}.$$

For strict monotonicity, fix $a := \Phi(\mathcal{Y}^{[b]};\mu) > 0$ and define $f(x) := \sqrt{a} - \sqrt{a-x}$ for $x \in [0,a]$. Since $f'(x) = 1/(2\sqrt{a-x}) > 0$, $f$ is strictly increasing on $[0,a]$, and $U_b(\mu) - U_b^{(s,a)}(\mu)$ is a positive constant multiple of $f(\Delta(s,a;\mathcal{Y}^{[b]},\mu))$. Hence $U_b(\mu) - U_b^{(s,a)}(\mu)$ is strictly increasing in $\Delta(s,a;\mathcal{Y}^{[b]},\mu)$. □

### C.4 Normalization of the reachability measure

Section 4.2 obtains the successor-guided distribution $\bar{H}_b$ by normalizing the positive part of the per-candidate scores $H_b(s_i,a_i)$ of equation (9) over the unlabeled pool. We make the normalization explicit. Let $\mathcal{U}_b := \mathcal{D} \setminus \mathcal{S}_b$ denote the unlabeled set at round $b$, and define

$$Z_b := \sum_{(s_j,a_j)\in\mathcal{U}_b} [H_b(s_j,a_j)]_+, \qquad [x]_+ := \max\{x,0\}.$$

The successor-guided distribution is

$$\bar{H}_b(s_i,a_i) := \begin{cases} \dfrac{[H_b(s_i,a_i)]_+}{Z_b}, & (s_i,a_i) \in \mathcal{U}_b \text{ and } Z_b > 0, \\ \dfrac{1}{|\mathcal{U}_b|}, & (s_i,a_i) \in \mathcal{U}_b \text{ and } Z_b = 0, \\ 0, & (s_i,a_i) \in \mathcal{S}_b. \end{cases}$$

Clipping to the positive part discards candidates whose successor overlap with the target set is negative under the current policy, so they receive no probability mass. The fallback to the uniform distribution on $\mathcal{U}_b$ when $Z_b = 0$—that is, when no unlabeled candidate has positive successor overlap with $T_b$—keeps $\bar{H}_b$ a well-defined probability measure on $\mathcal{U}_b$ at every round. Under this definition, the mixture $\rho_b = \beta_b \hat{d}_{\text{beh}}^{(b)} + (1-\beta_b)\bar{H}_b$ of equation (10) is always a probability measure on the unlabeled pool, with the behavior component controlling coverage and the successor component concentrating on candidates that can reach $T_b$ under $\pi^{[b]}$.

### C.5 Exact-gain scoring complexity

Section 4.2 states that, at each round $b$, evaluating the exact gain $\Delta(s_i,a_i;\mathcal{Y}^{[b]},\rho_b)$ of equation (8) for every unlabeled candidate costs $O(d^3 + (|\mathcal{D}| - b)d^2)$ when the directional operator is formed explicitly. We record the per-round cost breakdown.

By Proposition 1,

$$\Delta(s_i,a_i;\mathcal{Y}^{[b]},\rho_b) = \frac{\big\|\Sigma(\rho_b)^{1/2}\Lambda_b^{-1}\phi(s_i,a_i)\big\|_2^2}{1+\sigma_b(s_i,a_i)^2}.$$

A candidate-wise evaluation therefore needs:

1. the symmetric square root $\Sigma(\rho_b)^{1/2}$, via one eigendecomposition of the symmetric $d \times d$ matrix $\Sigma(\rho_b)$, at $O(d^3)$;

2. the matrix product $A_b := \Sigma(\rho_b)^{1/2}\Lambda_b^{-1} \in \mathbb{R}^{d\times d}$, at $O(d^3)$;
3. one matrix–vector product $A_b\phi(s_i, a_i)$ per unlabeled candidate, at $O(d^2)$, followed by an $O(d)$ squared-norm;
4. the scalar uncertainty $\sigma_b(s_i, a_i)^2 = \phi(s_i, a_i)^\top \Lambda_b^{-1}\phi(s_i, a_i)$, already maintained by the pessimistic reward pipeline.

Summing over the $|\mathcal{D}| - b$ unlabeled candidates, the per-round cost is $O(d^3 + (|\mathcal{D}| - b)d^2)$, as claimed. By contrast, the SURE score of equation (11) requires only $O((|\mathcal{D}| - b)d^2)$ per round to evaluate the $\sigma_b(s_i, a_i)^2$ values across the unlabeled pool—the $O(d^3)$ operator construction is avoided—while reusing the per-candidate scores $H_b(s_i, a_i)$ and the mixture weights $\rho_b(s_i, a_i)$ already produced by the surrogate-occupancy step.

# D Theoretical Analysis

This appendix supports the discussion at the end of Section 4.2. We define a coverage coefficient that measures the cost of replacing the unknown optimal occupancy $d^{\pi_r^\star}$ by a computable reference measure on the offline pool, prove the resulting unified suboptimality bound, derive a budget-dependent design rate that exposes the dependence on coverage mismatch and effective dimension, and explain how the SURE score arises as a scalable surrogate for the exact one-step gain. Every formal statement is proved.

## D.1 Coverage transfer and unified bound

**Definition 3** (Coverage coefficient and query complexity)**.** *For any probability measure $\nu$ on the unlabeled pool, define*

$$\kappa(\nu) := \inf\{c \geq 1 : \Sigma(d^{\pi_r^\star}) \preceq c\,\Sigma(\nu)\}, \qquad r_\nu := \operatorname{rank}(\Sigma(\nu)),$$

*and the query-complexity coefficient $\mathcal{C}_{\mathrm{qry}}(\nu) := \kappa(\nu)\, r_\nu$. We call $\nu$* admissible *if $\kappa(\nu) < \infty$. Admissibility is a data-coverage condition on the offline pool: $\kappa(\nu) < \infty$ requires $\operatorname{range}(\Sigma(d^{\pi_r^\star})) \subseteq \operatorname{range}(\Sigma(\nu))$. If the pool leaves any feature direction used by the optimal policy uncovered, no reward-labeling strategy can recover it from labels alone.*

**Proposition 2** (Coverage transfer)**.** *For any admissible reference measure $\nu$ and any labeled set $\mathcal{Y}$,*

$$\Phi(\mathcal{Y}; d^{\pi_r^\star}) \leq \kappa(\nu)\,\Phi(\mathcal{Y}; \nu).$$

*Proof.* By definition of $\kappa(\nu)$, $A := \kappa(\nu)\Sigma(\nu) - \Sigma(d^{\pi_r^\star}) \succeq 0$. Set $M := \Lambda(\mathcal{Y})^{-1} \succ 0$. For any pair of positive semidefinite matrices we have $\operatorname{tr}(AM) = \operatorname{tr}(M^{1/2}AM^{1/2}) \geq 0$, since $M^{1/2}AM^{1/2} \succeq 0$. Therefore

$$0 \leq \operatorname{tr}\big([\kappa(\nu)\Sigma(\nu) - \Sigma(d^{\pi_r^\star})]M\big) = \kappa(\nu)\operatorname{tr}(\Sigma(\nu)M) - \operatorname{tr}(\Sigma(d^{\pi_r^\star})M).$$

For any probability measure $\mu$, trace cyclicity together with $\sigma_b(s, a)^2 = \phi(s, a)^\top\Lambda(\mathcal{Y})^{-1}\phi(s, a)$ gives

$$\operatorname{tr}(\Sigma(\mu)M) = \operatorname{tr}\big(\mathbb{E}_{(s,a)\sim\mu}[\phi(s, a)\phi(s, a)^\top]\Lambda(\mathcal{Y})^{-1}\big) = \mathbb{E}_{(s,a)\sim\mu}\big[\phi(s, a)^\top\Lambda(\mathcal{Y})^{-1}\phi(s, a)\big] = \Phi(\mathcal{Y}; \mu).$$

Substituting $\mu = \nu$ and $\mu = d^{\pi_r^\star}$ yields $\Phi(\mathcal{Y}; d^{\pi_r^\star}) \leq \kappa(\nu)\,\Phi(\mathcal{Y}; \nu)$. □

**Proposition 3** (Coverage preserved by the behavior mixture)**.** *Let $\rho_b = \beta_b\hat{d}^{(b)}_{\mathrm{beh}} + (1 - \beta_b)\bar{H}_b$ be the round-$b$ reachability measure of equation (10). For any $\beta_b > 0$,*

$$\Sigma(\rho_b) \succeq \beta_b\,\Sigma(\hat{d}^{(b)}_{\mathrm{beh}}), \qquad \kappa(\rho_b) \leq \frac{\kappa(\hat{d}^{(b)}_{\mathrm{beh}})}{\beta_b}.$$

*This shows that successor guidance can concentrate mass near currently high-value targets without discarding the feature directions already covered by the empirical behavior distribution.*

*Proof.* By construction, $\bar{H}_b$ is a probability measure on the unlabeled pool (Appendix C.4), so its second-moment matrix is

$$\Sigma(\bar{H}_b) = \sum_i \bar{H}_b(s_i, a_i)\, \phi(s_i, a_i)\phi(s_i, a_i)^\top \succeq 0$$

as a conic combination of rank-one positive semidefinite matrices. Linearity of $\Sigma(\cdot)$ under convex combinations gives

$$\Sigma(\rho_b) = \beta_b\, \Sigma(\hat{d}^{(b)}_{\mathrm{beh}}) + (1-\beta_b)\, \Sigma(\bar{H}_b) \succeq \beta_b\, \Sigma(\hat{d}^{(b)}_{\mathrm{beh}}),$$

which is the first claim.

For the second claim, write $c_{\mathrm{beh}} := \kappa(\hat{d}^{(b)}_{\mathrm{beh}})$. By definition of $\kappa$, $\Sigma(d^{\pi^\star_r}) \preceq c_{\mathrm{beh}}\, \Sigma(\hat{d}^{(b)}_{\mathrm{beh}})$. The first claim and $\beta_b > 0$ imply $\Sigma(\hat{d}^{(b)}_{\mathrm{beh}}) \preceq \beta_b^{-1}\Sigma(\rho_b)$. Composing the two PSD inequalities,

$$\Sigma(d^{\pi^\star_r}) \preceq c_{\mathrm{beh}}\, \Sigma(\hat{d}^{(b)}_{\mathrm{beh}}) \preceq \frac{c_{\mathrm{beh}}}{\beta_b}\, \Sigma(\rho_b).$$

By minimality in the definition of $\kappa(\rho_b)$, $\kappa(\rho_b) \le c_{\mathrm{beh}}/\beta_b = \kappa(\hat{d}^{(b)}_{\mathrm{beh}})/\beta_b$. □

Combining Theorem 1 with Proposition 2 yields the structural bound under any admissible reference measure.

**Theorem 2** (Unified reference-measure suboptimality bound)**.** *Under Definition 1, equation (2), and equation (3), with probability at least $1-2\delta$, for any admissible reference measure $\nu$ and any labeled set $\mathcal{Y}$ producing a pessimistic policy $\hat{\pi}_{\tilde{r}}$ with confidence radius $\alpha$,*

$$\mathrm{SubOpt}(\mathcal{Q}^{(B)}) \le \Gamma_{\mathrm{off}}(\mathcal{D}, \delta) + \frac{2\alpha}{1-\gamma}\sqrt{\kappa(\nu)\, \Phi(\mathcal{Y}; \nu)}.$$

*Proof.* Theorem 1, applied to $\mathcal{Y}$ and the pessimistic policy $\hat{\pi}_{\tilde{r}}$ with confidence radius $\alpha$, gives

$$\mathrm{SubOpt}(\mathcal{Q}^{(B)}) \le \Gamma_{\mathrm{off}}(\mathcal{D}, \delta) + \frac{2\alpha}{1-\gamma}\sqrt{\mathbb{E}_{(s,a)\sim d^{\pi^\star_r}}[\sigma(s,a)^2]},$$

where $\sigma$ is the uncertainty radius associated with $\mathcal{Y}$. The trace-identity argument used in the proof of Proposition 2 (taking $\mu = d^{\pi^\star_r}$) yields $\mathbb{E}_{(s,a)\sim d^{\pi^\star_r}}[\sigma(s,a)^2] = \Phi(\mathcal{Y}; d^{\pi^\star_r})$. Substituting and applying Proposition 2,

$$\Phi(\mathcal{Y}; d^{\pi^\star_r}) \le \kappa(\nu)\, \Phi(\mathcal{Y}; \nu).$$

The square-root map is monotone increasing on $[0, \infty)$, so substituting yields the displayed bound. □

### D.2 Budget-dependent rates

We translate Theorem 2 into a budget-dependent rate by tracking how fast the design potential $\Phi(\mathcal{Y}^{[B]}; \nu)$ decays under exact-gain selection against a fixed reference measure.

**Theorem 3** (Fixed-measure greedy design rate)**.** *Fix an admissible probability measure $\nu$ on the pool with rank $r_\nu := \mathrm{rank}(\Sigma(\nu))$, and run the exact-gain rule*

$$(s_{i_b}, a_{i_b}) \in \arg\max_{(s_i,a_i)\in\mathcal{D}\setminus\mathcal{S}_b} \Delta(s_i, a_i; \mathcal{Y}^{[b]}, \nu)$$

*of Proposition 1 against this fixed reference measure. Then for every $B \ge 1$,*

$$\Phi(\mathcal{Y}^{[B]}; \nu) \le \frac{2\, r_\nu}{B}.$$

*Proof.* Write $F_b := \Phi(\mathcal{Y}^{[b]}; \nu) \ge 0$. By Proposition 1, $F_b$ is monotonically nonincreasing in $b$ and the round-$b$ drop equals the exact gain at the chosen candidate:

$$F_b - F_{b+1} = \Delta(s_{i_b}, a_{i_b}; \mathcal{Y}^{[b]}, \nu).$$

*Step 1: greedy lower bound.* The rule maximizes $\Delta(\cdot; \mathcal{Y}^{[b]}, \nu)$ over the unlabeled pool, and $\nu$ is supported on the pool, so

$$\Delta(s_{i_b}, a_{i_b}; \mathcal{Y}^{[b]}, \nu) \geq \mathbb{E}_{(s,a)\sim\nu}\big[\Delta(s, a; \mathcal{Y}^{[b]}, \nu)\big].$$

*Step 2: lower bound on the expected gain.* By Proposition 1,

$$\Delta(s, a; \mathcal{Y}^{[b]}, \nu) = \frac{\|\Sigma(\nu)^{1/2}\Lambda_b^{-1}\phi(s,a)\|_2^2}{1+\sigma_b(s,a)^2}.$$

Section 3 takes $\lambda \geq 1$ so that $\Lambda_b \succeq I$ and $\sigma_b(s,a)^2 = \phi(s,a)^\top \Lambda_b^{-1}\phi(s,a) \leq \|\phi(s,a)\|_2^2 \leq 1$. Hence the denominator is at most 2, and

$$\mathbb{E}_{(s,a)\sim\nu}\big[\Delta(s, a; \mathcal{Y}^{[b]}, \nu)\big] \geq \tfrac{1}{2}\,\mathbb{E}_{(s,a)\sim\nu}\big[\|\Sigma(\nu)^{1/2}\Lambda_b^{-1}\phi(s,a)\|_2^2\big].$$

Expanding the squared norm,

$$\|\Sigma(\nu)^{1/2}\Lambda_b^{-1}\phi(s,a)\|_2^2 = \phi(s,a)^\top\Lambda_b^{-1}\Sigma(\nu)\Lambda_b^{-1}\phi(s,a) = \mathrm{tr}\big(\Lambda_b^{-1}\Sigma(\nu)\Lambda_b^{-1}\phi(s,a)\phi(s,a)^\top\big),$$

using $x^\top M x = \mathrm{tr}(Mxx^\top)$. Taking expectation over $(s,a)\sim\nu$, with $\mathbb{E}_{(s,a)\sim\nu}[\phi(s,a)\phi(s,a)^\top] = \Sigma(\nu)$ and trace cyclicity,

$$\mathbb{E}_{(s,a)\sim\nu}\big[\|\Sigma(\nu)^{1/2}\Lambda_b^{-1}\phi(s,a)\|_2^2\big] = \mathrm{tr}\big(\Lambda_b^{-1}\Sigma(\nu)\Lambda_b^{-1}\Sigma(\nu)\big) = \mathrm{tr}(A_b^2) = \|A_b\|_F^2,$$

where $A_b := \Sigma(\nu)^{1/2}\Lambda_b^{-1}\Sigma(\nu)^{1/2} \succeq 0$.

*Step 3: relating $\|A_b\|_F^2$ to $F_b$.* Trace cyclicity gives $\mathrm{tr}(A_b) = \mathrm{tr}(\Lambda_b^{-1}\Sigma(\nu)) = F_b$ (the last equality is the trace identity from Step 2). The matrix $A_b$ has rank at most $\mathrm{rank}(\Sigma(\nu)) = r_\nu$, so its eigenvalues $\lambda_1(A_b), \ldots, \lambda_{r_\nu}(A_b) \geq 0$ (with the remaining $d - r_\nu$ eigenvalues equal to zero) satisfy, by the Cauchy–Schwarz inequality applied to the all-ones vector and $(\lambda_i)_{i\leq r_\nu}$,

$$F_b^2 = \Big(\sum_{i=1}^{r_\nu}\lambda_i(A_b)\Big)^2 \leq r_\nu \sum_{i=1}^{r_\nu}\lambda_i(A_b)^2 = r_\nu\,\|A_b\|_F^2.$$

Therefore $\|A_b\|_F^2 \geq F_b^2/r_\nu$, and combining Steps 1–3,

$$F_b - F_{b+1} \geq \frac{F_b^2}{2\,r_\nu}. \tag{21}$$

*Step 4: telescoping the recursion.* If $F_b = 0$ for some $b \leq B$, monotonicity gives $F_B = 0$ and the bound is trivial. Otherwise, dividing (21) by $F_bF_{b+1} > 0$ and using $F_{b+1} \leq F_b$,

$$\frac{1}{F_{b+1}} - \frac{1}{F_b} = \frac{F_b - F_{b+1}}{F_bF_{b+1}} \geq \frac{F_b - F_{b+1}}{F_b^2} \geq \frac{1}{2\,r_\nu}.$$

Summing over $b = 0, \ldots, B-1$ and using $F_0 \geq 0$ (so $1/F_0 \geq 0$),

$$\frac{1}{F_B} \geq \frac{1}{F_0} + \frac{B}{2\,r_\nu} \geq \frac{B}{2\,r_\nu},$$

which gives $F_B \leq 2\,r_\nu/B$, the claimed bound. □

Combining Theorem 2 with Theorem 3 yields the budget-dependent corollary that we view as the headline rate statement: under exact-gain selection against a fixed reference measure, the label-dependent term decays at the parametric rate $B^{-1/2}$ with a constant determined by the query complexity $\mathcal{C}_{\mathrm{qry}}(\nu) = \kappa(\nu)\,r_\nu$.

**Corollary 1** (Reference-measure suboptimality rate)**.** *Under the assumptions of Theorem 2 and exact-gain selection against an admissible fixed reference measure $\nu$,*

$$\mathrm{SubOpt}(\mathcal{Q}^{(B)}) \leq \Gamma_{\mathrm{off}}(\mathcal{D}, \delta) + \frac{2\alpha_B}{1-\gamma}\sqrt{\frac{2\,\mathcal{C}_{\mathrm{qry}}(\nu)}{B}}.$$

*Proof.* Theorem 2 applied at the terminal round $B$ with reference measure $\nu$ and confidence radius $\alpha = \alpha_B$ gives

$$\mathrm{SubOpt}(\mathcal{Q}^{(B)}) \leq \Gamma_{\mathrm{off}}(\mathcal{D}, \delta) + \frac{2\alpha_B}{1-\gamma}\sqrt{\kappa(\nu)\,\Phi(\mathcal{Y}^{[B]}; \nu)}.$$

By Theorem 3, $\Phi(\mathcal{Y}^{[B]}; \nu) \leq 2\,r_\nu/B$, and the square-root map is monotone, so

$$\sqrt{\kappa(\nu)\,\Phi(\mathcal{Y}^{[B]}; \nu)} \leq \sqrt{\kappa(\nu)\cdot\tfrac{2\,r_\nu}{B}} = \sqrt{\frac{2\,\mathcal{C}_{\mathrm{qry}}(\nu)}{B}},$$

using $\mathcal{C}_{\mathrm{qry}}(\nu) = \kappa(\nu)\,r_\nu$. Substituting yields the displayed bound. □

### D.3 Exact-gain structure and the SURE score

We finally explain why the SURE score (11) approximates the exact gain (8), and what is preserved when the directional operator $\Sigma(\mu)^{1/2}\Lambda_b^{-1}$ is replaced by the scalar mixture weight $\rho_b$.

For a fixed round $b$, reference measure $\mu$, and candidate $(s,a)$, define

$$v_b(s,a) := \Lambda_b^{-1}\phi(s,a), \qquad \varrho_b(s,a)^2 := \|v_b(s,a)\|_2^2, \qquad u_b(s,a) := v_b(s,a)/\|v_b(s,a)\|_2 \quad (\text{when } v_b(s,a) \neq 0).$$

Define the directional relevance and the uncertainty-reduction factor

$$\widetilde{w}_b(s,a;\mu) := u_b(s,a)^\top \Sigma(\mu) u_b(s,a), \qquad \zeta_b(s,a) := \frac{\varrho_b(s,a)^2}{1+\sigma_b(s,a)^2}.$$

When $v_b(s,a) = 0$, set both quantities to zero.

**Proposition 4** (Exact-gain decomposition)**.** *For every round $b$, every reference measure $\mu$, and every candidate $(s,a)$,*

$$\Delta(s,a;\mathcal{Y}^{[b]},\mu) = \widetilde{w}_b(s,a;\mu)\,\zeta_b(s,a).$$

*Proof.* By Proposition 1,

$$\Delta(s,a;\mathcal{Y}^{[b]},\mu) = \frac{\|\Sigma(\mu)^{1/2}\Lambda_b^{-1}\phi(s,a)\|_2^2}{1+\sigma_b(s,a)^2}.$$

For any vector $v \in \mathbb{R}^d$ and PSD $\Sigma$, $\|\Sigma^{1/2}v\|_2^2 = v^\top \Sigma v$. Applying this with $v = v_b(s,a) = \Lambda_b^{-1}\phi(s,a)$ and $\Sigma = \Sigma(\mu)$,

$$\|\Sigma(\mu)^{1/2}\Lambda_b^{-1}\phi(s,a)\|_2^2 = v_b(s,a)^\top \Sigma(\mu) v_b(s,a).$$

If $v_b(s,a) = 0$, the right-hand side is zero, so $\Delta(s,a;\mathcal{Y}^{[b]},\mu) = 0$; by definition $\widetilde{w}_b(s,a;\mu)\,\zeta_b(s,a) = 0$ as well, and the identity holds trivially. Otherwise $v_b(s,a) = \varrho_b(s,a)\,u_b(s,a)$ and

$$v_b(s,a)^\top \Sigma(\mu) v_b(s,a) = \varrho_b(s,a)^2\, u_b(s,a)^\top \Sigma(\mu) u_b(s,a) = \varrho_b(s,a)^2\, \widetilde{w}_b(s,a;\mu).$$

Substituting,

$$\Delta(s,a;\mathcal{Y}^{[b]},\mu) = \frac{\varrho_b(s,a)^2\,\widetilde{w}_b(s,a;\mu)}{1+\sigma_b(s,a)^2} = \widetilde{w}_b(s,a;\mu)\,\zeta_b(s,a),$$

which is the claimed factorization. □

The SURE score (11) replaces the two factors of Proposition 4 by quantities already produced upstream. The directional relevance $\widetilde{w}_b(\cdot;\mu)$ is replaced by the scalar mixture weight $\rho_b(s,a)$ – which reuses the reachability scores of equation (9) and the behavior mixing of equation (10) – and the uncertainty-reduction factor $\zeta_b$ is replaced by the closely related $\sigma_b(s,a)^2/(1+\sigma_b(s,a)^2)$, already maintained by the pessimistic reward pipeline. The next proposition records the cost saving that this substitution buys per round.

**Proposition 5** (Per-round cost of the SURE score versus the exact gain)**.** *At each round $b$, evaluating $\mathrm{SURE}_b$ of equation (11) for every unlabeled candidate costs $O((|\mathcal{D}|-b)\,d^2)$, whereas evaluating the exact gain $\Delta(\,\cdot\,;\mathcal{Y}^{[b]},\rho_b)$ of equation (8) for every unlabeled candidate costs $O(d^3 + (|\mathcal{D}|-b)\,d^2)$ when the directional operator $\Sigma(\rho_b)^{1/2}\Lambda_b^{-1}$ is formed explicitly.*

*Proof.* The SURE score requires only the per-candidate scalars $\sigma_b(s_i,a_i)^2 = \phi(s_i,a_i)^\top \Lambda_b^{-1}\phi(s_i,a_i)$ and the mixture weights $\rho_b(s_i,a_i)$. Once $\Lambda_b^{-1}$ is maintained incrementally by the pessimistic reward pipeline, each $\sigma_b(s_i,a_i)^2$ costs one matrix–vector product and one inner product, i.e., $O(d^2)$; summed over the $|\mathcal{D}|-b$ unlabeled candidates this gives $O((|\mathcal{D}|-b)\,d^2)$.

For the exact gain, Proposition 1 requires the directional operator $A_b := \Sigma(\rho_b)^{1/2}\Lambda_b^{-1}$. Computing the symmetric square root $\Sigma(\rho_b)^{1/2}$ needs one eigendecomposition of the symmetric $d\times d$ matrix $\Sigma(\rho_b)$, at $O(d^3)$, and forming $A_b$ needs one $d\times d$ matrix product, also at $O(d^3)$. Then for each unlabeled candidate, computing $A_b\phi(s_i,a_i)$ costs $O(d^2)$ and the squared norm an additional $O(d)$. Summing the operator-construction cost and the $|\mathcal{D}|-b$ per-candidate matrix–vector products gives $O(d^3 + (|\mathcal{D}|-b)\,d^2)$ per round, as claimed. □

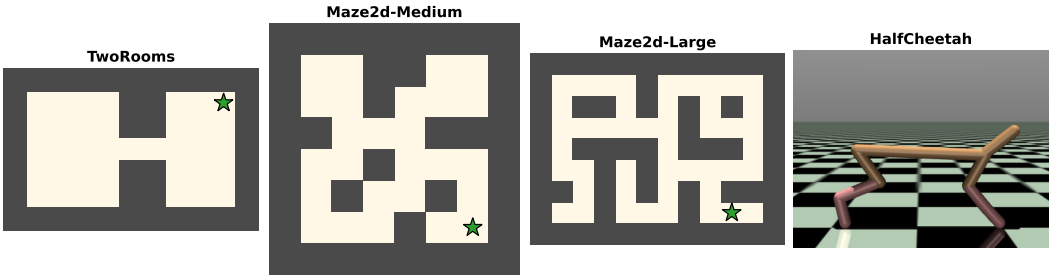


Figure 4: Visualization of the four evaluation domains. From left to right: TwoRooms, Maze2d-Medium, Maze2d-Large, and HalfCheetah-Medium-Replay.

SURE thus preserves the relevance-times-uncertainty structure of the exact gain – each candidate is scored by the product of a relevance term and an uncertainty term – while avoiding the per-round $O(d^3)$ operator construction. This is the structural reason why dropping either factor (REACH or UNCERTAINTY in the empirical ablation of Section 5.3) breaks the score whenever the dropped factor is not nearly flat across candidates.

# E Experimental Details

This appendix complements Section 5 with domain and dataset details, baseline implementations, the shared partially labeled offline-RL pipeline, and acquisition hyperparameters.

## E.1 Domains and Datasets

We evaluate on four offline-RL domains chosen to span continuous navigation, continuous-control navigation at two scales, and high-dimensional locomotion. Within each domain, every reward-selection strategy and every random seed sees the same fixed reward-free transition pool $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^{|\mathcal{D}|}$; a strategy is allowed to acquire $B$ transition-level reward labels from this pool.

### E.1.1 Domain visualizations

Figure 4 shows one subplot per domain. The four subplots correspond to the following.

- **TwoRooms.** A continuous two-room gridworld in which two rectangular rooms are connected by a narrow doorway. The state is the agent's continuous cell and the action set is the four cardinal moves; a sparse reward is given only at the goal cell, which lies in the room opposite the start. The doorway concentrates the directions an optimal policy must traverse, providing a small-scale sanity check for the reward-selection problem.
- **Maze2d-Medium.** A continuous 2D point-mass navigation domain from the D4RL benchmark suite [36]. The state encodes the point-mass position and velocity, the action is a continuous 2D force, and a sparse reward is given upon reaching the goal. "Medium" refers to the layout size: a moderately long path between start and goal with a small number of corridors.
- **Maze2d-Large.** The larger-layout counterpart of Maze2d-Medium, with the same continuous state and action spaces and the same sparse goal-reaching reward, but a longer required path between start and goal. The longer horizon concentrates the optimal-policy directions on a narrow corridor that occupies a smaller fraction of the offline pool, increasing the difficulty of credit assignment.
- **HalfCheetah-Medium-Replay.** A continuous-control locomotion domain from the D4RL MuJoCo suite [35, 36]. The state encodes joint positions and velocities of the simulated half-cheetah (17-D), and the action is a continuous torque vector (6-D). The Medium-Replay dataset records the full replay buffer of a partially trained agent, so it mixes high- and low-quality transitions; together with the navigation domains it provides a higher-dimensional, non-navigation contrast for the acquisition comparison.

### E.1.2 Dataset collection

We use the same fixed reward-free transition pool $\mathcal{D}$ for every reward-selection strategy and every random seed within a domain.

**TwoRooms.** We collect the offline pool ourselves, with $|\mathcal{D}| = 15{,}000$ transitions. The data-collecting policy mixes uniform random exploration with goal-directed trajectories from a hand-coded shortest-path policy: at each step it takes a random action with probability $0.5$ and a shortest-path action with probability $0.5$. This mixture ensures coverage of both the corridor connecting the two rooms and the surrounding non-corridor cells.

**Maze2d-Medium and Maze2d-Large.** We use the standard D4RL datasets "maze2d-medium-v1" ($|\mathcal{D}| \approx 2$M) and "maze2d-large-v1" ($|\mathcal{D}| \approx 4$M), both collected by a hand-engineered planner that drives the point mass to repeatedly resampled goals. We strip the rewards from the released datasets to obtain the reward-free pool used as input to reward selection.

**HalfCheetah-Medium-Replay.** We use the D4RL dataset "halfcheetah-medium-replay-v2" ($|\mathcal{D}| \approx 200$K), which records the full replay buffer of a SAC agent trained to medium-level performance. Rewards are stripped to obtain the reward-free pool used as input to reward selection.

## E.2 Retained-Label Reward Selection Baselines

All baselines share the offline pool $\mathcal{D}$ and the same partially labeled offline-RL pipeline of Appendix E.3; they differ only in how each round's transition is selected from the unlabeled pool. We describe the implementation of each in turn.

**Uniform.** At round $b$, Uniform draws a transition uniformly at random from the unlabeled subset $\mathcal{D} \setminus \mathcal{S}_b$, without replacement across rounds. Uniform uses no value information, no reward-uncertainty information, and no policy estimate, and serves as the coverage-only baseline.

**RLLF-Guided.** RLLF-Guided is the guided heuristic of Chaudhari et al. [9]. At round $b$, with $Q_b$ the action-value function of the current pessimistic policy $\pi^{[b]}$ under reward $\tilde{r}_b$ and $\hat{z}_b^\star \in \arg\max_{(s,a,s')\in\mathcal{D}} Q_b(s,a)$ the current top-valued transition, each unlabeled candidate $(s_i, a_i)$ is scored by an empirical one-step transition probability of reaching $\hat{z}_b^\star$ from $(s_i, a_i)$, estimated empirically from the offline transition pool. The score is then mixed with the empirical behavior visitation distribution as in Chaudhari et al. [9], and the candidate with the highest mixed score is selected. We re-run this entire procedure at every round.

**IDRL.** IDRL is the Information Directed Reward Learning method of Lindner et al. [13], originally proposed for online reward querying. At each round it selects the candidate label that is expected to be most informative for comparing the returns of candidate high-performing policies. To adapt IDRL to our offline setting, we replace its online rollout-based return estimation with FQE [38] on the fixed dataset $\mathcal{D}$, so that policy-return estimates used to compute the information-gain criterion are derived from the same offline pool used by every other baseline. The candidate set, the posterior reward distribution, and the FQE evaluation are recomputed at every round.

**Reach (single-factor ablation).** Reach is a single-factor ablation of SURE that drops the reward-uncertainty term. At round $b$, it selects the candidate $(s_i, a_i) \in \mathcal{D} \setminus \mathcal{S}_b$ that maximizes the relevance weight $\rho_b(s_i, a_i)$ from equation (10). Reach uses the same successor-feature reachability and behavior-mixing pipeline as SURE, but ignores how much each label would shrink the reward-confidence radius.

**Uncertainty (single-factor ablation).** Uncertainty is the complementary single-factor ablation: it drops the relevance weight and selects the candidate that maximizes $\sigma_b(s_i, a_i)^2/(1 + \sigma_b(s_i, a_i)^2)$, using the current pessimistic-reward design matrix $\Lambda_b$. Uncertainty uses no policy or successor information; it queries wherever the reward model is currently most uncertain.

### E.3 Partially Labeled Offline RL Implementation

After a reward-selection strategy produces the retained labeled set $\mathcal{Y}^{[b]}$, every method follows the same pipeline from Section 3: the retained labels are used to construct a conservative reward estimate, the reward-free transitions in $\mathcal{D}$ are annotated with this estimate, and IQL is trained on the resulting dataset. This shared pipeline ensures that empirical differences come from the reward-selection rule rather than the offline-RL learner. All methods use the same representation and offline-RL implementation.

The IQL learner [37] hyperparameters are listed in Table 2; we follow standard IQL settings throughout.

Table 2: IQL hyperparameters.

| Hyperparameter | Value |
|---|---|
| Hidden sizes for $V$, $Q$, $\pi$ | $256 \times 256$ MLP, ReLU |
| Optimizer | Adam |
| Learning rate | $3 \times 10^{-4}$ (all three networks) |
| Batch size | 256 (D4RL); 64 (TwoRooms) |
| Discount $\gamma$ | 0.99 (HC-MR); 0.995 (Maze2d-M, Maze2d-L); 0.95 (TwoRooms) |
| Target-network Polyak rate $\tau_{\text{soft}}$ | $5 \times 10^{-3}$ |
| Expectile parameter $\tau$ | 0.7 (D4RL); 0.5 (TwoRooms) |
| Advantage-weighting temperature $\beta$ | 3.0 (Maze2d-M, HC-MR); 10.0 (Maze2d-L); 1.0 (TwoRooms) |
| Reward scaling | fixed across methods |

### E.4 Hyperparameters

**SURE acquisition score.** The acquisition-score hyperparameters are listed in Table 3. We set the top-$K$ target-set size $|T_b|$ by domain scale, using a smaller $K$ for TwoRooms and a larger $K$ for the continuous domains. We use a small constant behavior-mixing weight $\beta_b = 0.1$ in all experiments; this prevents the relevance weight from concentrating on isolated outliers under low-coverage early rounds.

Table 3: SURE acquisition-score hyperparameters.

| Hyperparameter | Value |
|---|---|
| Top-$K$ target-set size $\lvert T_b \rvert$ | 5 (TwoRooms); 32 (Maze2d-M); 64 (Maze2d-L); 64 (HC-MR) |
| Behavior-mixing weight $\beta_b$ | 0.1 (constant across rounds) |
| Successor-feature ridge regularizer $\lambda_{\text{SF}}$ | $10^{-3}$ |
| Successor-feature estimator | closed-form ridge (Appendix B.2) |
| Refit cadence | every round (single-label acquisition) |

### E.5 Evaluation Protocol

We report mean return over 50 random seeds; shaded regions in Figure 2 show standard error. Policies are evaluated with 100 rollouts per seed for the D4RL domains and 200 rollouts for TwoRooms. All methods are evaluated under the same budget grid and the same full-feedback reference. Due to hardware constraints, we run our jobs on CPUs.

## F Limitations

Our theoretical analysis is stated under the linear-MDP model of Definition 1 and presumes that the chosen reference measure is admissible (Definition 3); when the offline pool fails to cover feature directions used by the optimal policy, no reward-labeling strategy can recover them.