Uncertainty-Guided Reward Labeling for Reinforcement Learning under Limited Feedback

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

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

Bruno Castro da Silva
University of Massachusetts
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 high-performing 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 in terms of two sources of error: one arising from offline RL on a fixed, offline dataset, and the other from uncertainty in the reward model. 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 introduce *Successor-Guided Uncertainty REduction* (SURE), a novel method that uses successor features to select rewards that are both uncertain and associated with states from which high-valued states are reachable. We formally 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 and offline-RL backbones, yielding an effective 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 [8, 28, 2]; in drug discovery settings, generative models can propose large candidate sets, while validation requires expensive simulation or experimental evaluation [32, 14, 17, 31, 11]; and in offline RL, large reward-free transition datasets may be available while reward annotation remains limited [7, 40, 15]. 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 training [7]. 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 [40, 15, 23]. However, these methods assume that the labeled subset is passively

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

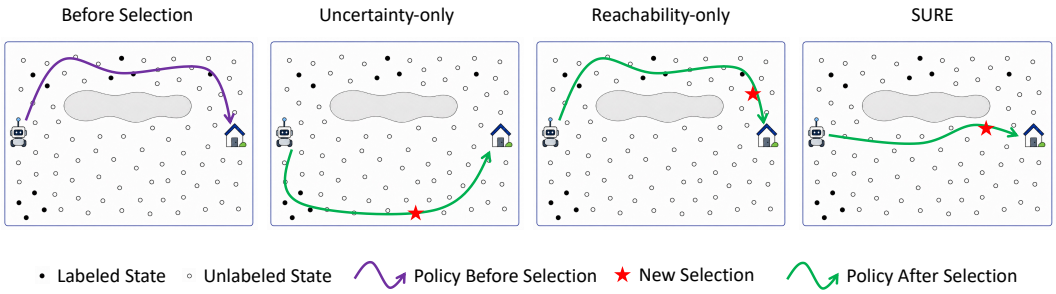

Figure 1: Illustrative retained-label reward-selection example in a navigation task. The agent must reach the goal while avoiding a central obstacle. With the initial reward labels, the policy learned from the current reward model takes a long detour, while a more direct path remains possible. Uncertainty-only 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 for labeling 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.

available. They do not address the question of which additional transitions should be selected for reward labeling. Information-directed reward learning (IDRL) is the closest related approach, as it actively queries feedback to learn a reward model for policy optimization. However, it acquires feedback online to distinguish the returns of candidate high-performing policies, rather than to determine which transitions in an offline dataset should be reward-labeled [26]. This leaves open the problem of *retained-label reward selection*: actively selecting a small subset of transitions from a fixed, offline dataset for reward labeling, so that a reward model trained on those labels can annotate the remaining transitions and enable learning a high-performing policy from the resulting dataset.

In this retained-label setting, selected rewards affect policy learning through the reward model they induce. A natural way to improve such a model is to label rewards that reduce uncertainty in the reward model, but reducing uncertainty at different states can have very different effects on the performance of 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 state occupancy of an optimal policy were known, the rule would be simple: select rewards that reduce uncertainty in states with high occupancy under that policy. Since this information is unavailable during selection, we introduce *Successor-Guided Uncertainty REduction* (SURE): the policy learned from the current reward model provides an evolving estimate of which states are valuable, while successor features are used to identify candidate states from which those states are reachable. SURE selects rewards where reachability and uncertainty meet, i.e., 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. Our analyses relate selection quality to coverage mismatch and effective dimension. Experiments across a variety of domains and offline-RL backbones 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 involve learners that can query different objects. Active reward learning and preference-based RL query comparisons, demonstrations, rankings, or other types of human feedback to identify a latent reward function [9, 34, 4, 38]. Information-directed reward learning (IDRL) is closer to our motivation: these methods actively acquire feedback to learn a reward model for policy optimization [26]. However, IDRL acquires feedback online, selecting queries expected to rapidly identify which candidate high-performing policy achieves the highest return, whereas our setting focuses on selecting which transitions in a fixed offline dataset should receive reward labels. Methods for active RL with costly rewards and partially observable reward models also query reward observations while interacting with the environment, intertwining reward acquisition with exploration and data collection [21, 35, 37, 30, 29]. 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 [6, 7], 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 signals guide 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 [40]. Another strategy is to infer missing reward signals from the labeled subset and use the learned reward model to annotate reward-free transitions before applying standard offline RL [19, 45, 33]. 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 [15]. Kernelized variants extend this partially labeled offline-RL framework beyond the original linear setting [23]. 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 [25, 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 [39, 16])**.** *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.*

Given 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. Consider this MDP, but now with a different reward function r'; we denote π's discounted expected return under r' 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 π 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 $J_r(\pi) = (1 - \gamma)^{-1} \mathbb{E}_{(S,A) \sim d^\pi}[r(S, A)]$ [36]. A closely related concept is *successor features*, which encode the expected discounted occupancy of future state-action features.

Definition 2 (Successor Features [10, 3])**.** *For a policy π, the successor features of a state–action pair (s, a) is defined as:*

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

Successor features represent the discounted future occupancy of the feature ϕ under a given 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. [7] (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 π_D with $\mathcal{M}$, and contains no rewards. Let $\mathcal{S}_{[B]}$ denote the states in $\mathcal{D}$ 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 states in $\mathcal{S}_{[B]}$ are not retained, in our setting each transition $(s_i, a_i, s'_i) \in \mathcal{D}$, where $s_i \in \mathcal{S}_{[B]}$, 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. In our setting, the selected rewards are retained and used to learn a policy $\pi^{[B]}$ through the reward-modeling and offline-RL pipeline described next. Our reward-selection objective, in particular, is to identify a querying 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 (PDS) framework [15]: fit a conservative reward model from $\mathcal{Y}$, use it to assign rewards to the remaining unlabeled 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 ℓ_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. [1] and the nonnegative-reward clipping in Definition 1, with probability at least $1 - \delta$, the following holds simultaneously for all labeled sets $\mathcal{Y}$ encountered by the learner and for 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 [18] as a black box providing the following high-probability guarantee: for any pessimistic reward estimate $\tilde{r} \leq r$,

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

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 δ, not on which transitions are reward-labeled. This motivates the reward-selection question studied next: which rewards should be queried/labeled (and retained) so that the resulting reward model improves policy learning the most?

4 Successor-Guided Uncertainty Reduction

We introduce *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 dataset after each queried reward, refits a pessimistic estimate of the reward model, and uses the resulting policy and value estimates to choose the next label. In round $b \in \{0, \dots, B-1\}$, SURE fits the current pessimistic reward model $\tilde{r}_b$ from $\mathcal{Y}^{[b]}$, runs the offline-RL learner to obtain $\pi^{[b]}$, selects an unlabeled 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 guiding this selection is derived below from a *label-sensitive bound* (Thm. 1) and then approximated through successor-guided reachability and reward-model uncertainty metrics (Section 4.2).

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 contribution of the fixed offline transition dataset from the contribution of the rewards selected so far.

Theorem 1 (Label-sensitive suboptimality decomposition)**.** *Assume that the MDP is linear, as in Definition 1, that the retained labels induce a conservative reward estimate satisfying the reward-confidence bound (Eq. (2)), and that the pessimistic offline-RL algorithm satisfies the stated offline-learning guarantee (Eq. (3)). For any strategy with budget B, let $\mathcal{Q}^{(b)}$ denote its first b selections, let $\mathcal{Y}^{[b]}$ denote the retained labeled set resulting from 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 $b \in \{0, \dots, B\}$,*

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

where $\mathrm{SubOpt}(\mathcal{Q}^{(b)}) := J_r(\pi_r^\star) - J_r(\pi^{[b]})$ 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. Thm. 1 shows that the suboptimality bound for the policy induced by a querying strategy $\mathcal{Q}^{(b)}$ can be tightened by minimizing a reward-uncertainty term that depends on the optimal occupancy $d^{\pi_r^\star}$ under the true reward function. Because $d^{\pi_r^\star}$ is unknown during selection, we need a computable surrogate (see Section Section 4.2). Here, we first derive the corresponding selection objective for an arbitrary reference measure μ.

Given a retained labeled set $\mathcal{Y}^{[b]}$ and reference measure μ, define the *weighted uncertainty objective* as

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

This objective measures the squared reward uncertainty averaged over μ. Thm. 1 can then be written as

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

Since only the second term in Eq. (6) depends on the selected rewards, we can use $\Phi(\mathcal{Y}^{[b]}; \mu)$ as the objective guiding reward selection. For a fixed reference measure μ, we define the **one-step acquisition function** as the uncertainty reduction in this objective obtained by reward-labeling a candidate (s, a):

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

We now show (Prop. 1) that this marginal gain admits a closed form under the linear-MDP assumption.

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 μ 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 Δ accounts for both the reference measure and the current labeled set: it favors candidates that reduce uncertainty in directions emphasized by μ, while downweighting feature directions already well represented by the current labels. For fixed μ, 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 $\hat{d}^{(b)}_{\mathrm{beh}}$ over unlabeled samples:

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

We refer to ρ_b as the successor-guided reachability measure.

The measure ρ_b is the computable reference measure used in place of the unavailable optimal occupancy $d^{\pi^\star_r}$. 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 ρ_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 ρ_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

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

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 5.4 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.

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 five questions: (**Q1** Section 5.1) when the linear-MDP assumptions of Section 3 hold by construction, does the computable SURE score retain the label efficiency of the exact one-step-gain rule it approximates; (**Q2** Section 5.2) does SURE remain label-efficient beyond

this setting, in continuous-control domains where the feature map is learned and the linear-MDP structure need not hold, and how does it compare with other reward-selection strategies across budgets; (**Q3** Section 5.3) how much feedback each strategy requires to approach the full-feedback performance obtained when rewards are available for all transitions in $\mathcal{D}$; (**Q4** Section 5.4) how each component of SURE contributes to acquisition; and (**Q5** Section 5.5) whether the gains of SURE are specific to the IQL backbone or persist when the downstream offline-RL learner is replaced.

We evaluate on five domains with different acquisition challenges: LinearMDP, a finite linear MDP in which the assumptions of Section 3 hold by construction (Section 5.1); TwoRooms, Maze2d-Medium, and Maze2d-Large, three navigation tasks with bottleneck and goal-reaching rewards; and HalfCheetah, a high-dimensional locomotion task [5]. 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 [12] 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 [7], which biases selection toward one-step predecessors of states estimated to have high value under the current learner. IDRL [26] 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 [20] is then run on the resulting reward-labeled dataset to produce a policy (Section 5.5 replaces IQL with other offline-RL backbones).

We report the return of learned policies as a function of the absolute feedback budget B in Section 5.2 and the percentage feedback $B/|\mathcal{S}_\mathcal{D}|$ in Section 5.3, 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 Validation under Exact Linear MDPs

We first evaluate SURE in a setting where the theoretical assumptions hold exactly. We construct finite linear MDPs using the simplex-feature construction of Jin et al. [16]: each state–action pair is assigned a known 8-dimensional feature vector on the probability simplex, its transition distribution is the corresponding convex combination of eight base next-state distributions, and its reward is linear in the same feature vector. Both the reward and the transition kernel are therefore linear in a fixed feature map ϕ, which is provided to every method. We consider environments with 32 and 48 states, four actions, and low-exploration offline pools of 4,096 transitions. Because ϕ is known and the problem is small, the exact one-step gain of Proposition 1 can be computed for every candidate, so we compare SURE with EXACT GAIN, which selects the candidate maximizing equation (8), and with UNIFORM. All methods share the same four uniformly selected warm-start labels, counted toward the budget, and results are averaged over 30 paired seeds.

Table 1 shows that SURE and EXACT GAIN require identical numbers of labels at every threshold in both environments. With 48 states, both reach 90% and 95% of the full-feedback return with 12 labels, whereas UNIFORM requires 68 and 132 labels, respectively. Thus, when the theoretical assumptions hold, replacing the exact one-step gain with the computable SURE score does not sacrifice label efficiency.

Table 1: Number of reward labels required to reach 80% / 90% / 95% of the matched full-feedback return in exact linear MDPs, averaged over 30 paired seeds. Full feedback corresponds to labeling all 4,096 transitions in the offline pool.

Environment	SURE	EXACT GAIN	UNIFORM
$\lvert\mathcal{S}\rvert = 32$	**8 / 12 / 12**	**8 / 12 / 12**	20 / 20 / 36
$\lvert\mathcal{S}\rvert = 48$	**8 / 12 / 12**	**8 / 12 / 12**	12 / 68 / 132

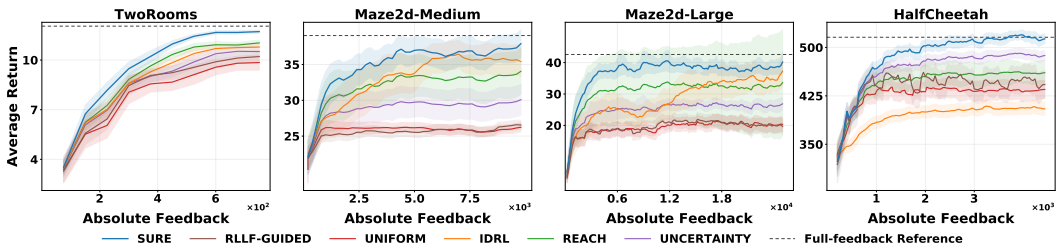

Figure 2: Performance comparison across domains under limited feedback. Return is shown as a function of the absolute feedback budget B. We compare SURE with UNIFORM, REACH, UNCERTAINTY, IDRL, and RLLF-GUIDED. Solid lines show the mean over 50 trials and shaded regions show the standard error. The dashed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

5.2 Label Efficiency Across Domains

In continuous-control tasks, however, a feature map under which rewards and transitions are exactly linear is generally unknown and must be learned from data. We therefore test whether the conclusions of Section 5.1 carry over to four continuous-control domains in which ϕ is learned rather than given. 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:

- **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.

5.3 Feedback Needed to Reach Full-Feedback Performance

Table 2 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

Table 2: 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%

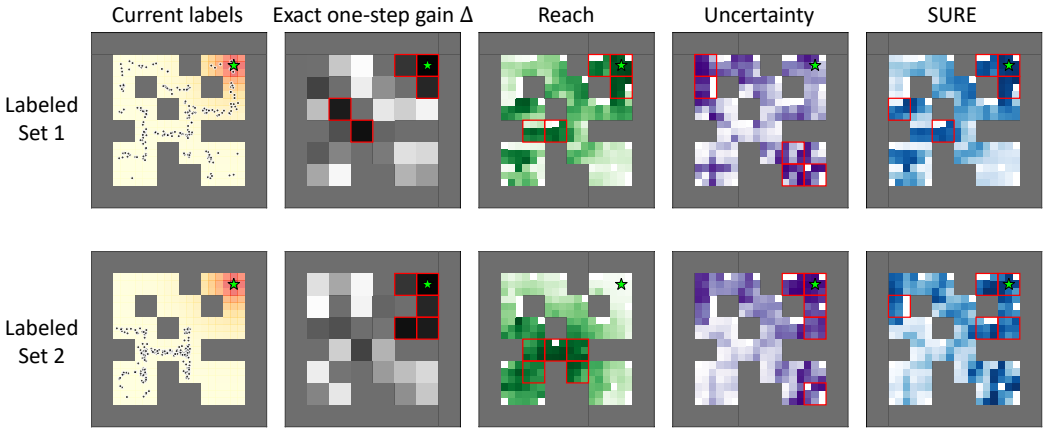

Figure 3: Ablation of the SURE score on representative maze rounds. Each row shows a different stage of selection. From left to right: acquired feedback locations, exact one-step gain $\Delta(s, a; \mathcal{Y}_{[b]}, \rho_b)$, Reach-only score, Uncertainty-only score, and SURE score. Red boxes mark the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

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 improve with larger budgets, but Table 2 shows that they require much more feedback to reach the same performance levels.

5.4 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

Table 3: Percentage feedback required to reach 90% of each backbone's own matched full-feedback reference, averaged over 50 paired seeds.

Backbone	Domain	SURE	UNIFORM
TD3+BC	Maze2d-Medium	**0.14%**	1.92%
	HalfCheetah	**0.52%**	2.36%
CQL	Maze2d-Medium	**0.16%**	2.00%
	HalfCheetah	**0.55%**	2.44%

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.

5.5 Robustness to the Offline-RL Backbone

All preceding experiments use the same IQL backbone for every method to isolate differences due to reward-label acquisition. However, SURE is not intrinsically tied to IQL: the analysis in Section 4 invokes the downstream learner only through the pessimistic offline-learning condition in equation (3). To test whether the empirical gains are specific to IQL, we replace the backbone throughout the full acquisition loop, including the policy and value estimates used to construct the SURE score, with CQL [22] and TD3+BC [13] on Maze2d-Medium and HalfCheetah. CQL provides a more theory-aligned test because it explicitly encourages conservative value estimates, whereas TD3+BC tests the acquisition rule with a behavior-regularized learner outside the formal condition.

Table 3 shows that SURE requires approximately $4.4\times$–$13.7\times$ less feedback than UNIFORM under both alternative backbones and in both domains, indicating that the acquisition benefit of SURE is not specific to IQL. The sensitivity of SURE to the size of the initial warm-start set is reported in Appendix E.7.

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 a variety of domains and offline-RL backbones, while other selection strategies are strongest mainly in regimes favored by their respective biases.

Acknowledgments and Disclosure of Funding

This work used computational resources and services provided by the Unity Research Computing Platform and the Swarm cluster at the University of Massachusetts Amherst. We also thank the anonymous reviewers for their comments and suggestions, which helped strengthen and clarify this work.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] Shreyas Chaudhari, Renhao Zhang, Philip S. Thomas, and Bruno Castro da Silva. Data selection for reward labeling in limited-feedback reinforcement learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=N8OlHGeHSy.

[8] 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.

[9] 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.

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

[11] 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.

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

[13] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

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

[22] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

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

[30] 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.

[31] 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.

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

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

[34] 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.

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

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

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

[38] 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.

[39] 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.

[40] 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.

[41] Renhao Zhang, Haotian Fu, Yilin Miao, and George Konidaris. Model-based reinforcement learning for parameterized action spaces. In *International Conference on Machine Learning*, 2024.

[42] Renhao Zhang, Haotian Fu, Mingxi Jia, George Konidaris, Yilun Du, and Bruno Castro da Silva. From noise to control: Parameterized diffusion policies. In *International Conference on Machine Learning*, 2026.

[43] Renhao Zhang, Mingzhe Li, and Bruno Castro da Silva. Latent barrier steering: Hierarchical safety for generative planning. In *Advances in Neural Information Processing Systems*, 2026.

[44] Renhao Zhang, Mingzhe Li, Yuli Li, Jiayu Zheng, and Yilin Miao. Learning what to keep: Evidence-gated parameterized skills for frozen LLM agents. In *NeurIPS 2026 Workshop on Test-Time Continual Learning*, 2026.

[45] 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.

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. Section 4 uses single-query notation to simplify the presentation of the acquisition rule and its analysis; in the experiments, labels are acquired in batches. SURE starts from B_0 uniformly sampled warm-start labels, which are shared by all methods within each seed and counted toward the feedback budget B. Each subsequent round b acquires the M unlabeled candidates with the largest SURE scores, after which the reward model, the offline backbone, the successor features, and the acquisition scores are recomputed. With $B_0 = 0$ and $M = 1$, round b coincides with the b-th query and Algorithm 1 reduces to the single-query rule of Section 4. We use $B_0 = M = 64$ in all continuous-control experiments (Appendix E.4). After the budget is exhausted, SURE refits the reward model using the final labeled set $\mathcal{Y}^{[R]}$, re-runs the pessimistic backbone once more on the full dataset, and returns the resulting policy $\pi^{[R]}$.

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

Require: offline dataset $\mathcal{D}$; budget B; warm-start size B_0; batch size M; top-K; weights $\{\beta_b\}$.

Draw $\mathcal{S}_0 \subset \mathcal{D}$ with $|\mathcal{S}_0| = B_0$ uniformly at random; query their rewards to form $\mathcal{Y}^{[0]}$.
Set the number of acquisition rounds $R \leftarrow (B - B_0)/M$.
for $b = 0, 1, \ldots, R-1$ **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: $\rho_b = \beta_b \hat{d}^{(b)}_{\mathrm{beh}} + (1 - \beta_b)\bar{H}_b$.
    $U_b \leftarrow$ the M candidates in $\mathcal{D} \setminus \mathcal{S}_b$ with the largest $\mathrm{SURE}_b(s_i, a_i)$ in (11).
    Query $r_i = r(s_i, a_i)$ for all $(s_i, a_i, s'_i) \in U_b$; update $\mathcal{Y}^{[b+1]} \leftarrow \mathcal{Y}^{[b]} \cup \{(s_i, a_i, r_i, s'_i)\}_{(s_i,a_i,s'_i)\in U_b}$, $\mathcal{S}_{b+1} \leftarrow \mathcal{S}_b \cup U_b$.
end for
Refit $\tilde{r}_R$ from $\mathcal{Y}^{[R]}$; run backbone on $\mathcal{D}$ with $\tilde{r}_R$ to obtain $\pi^{[R]}$.
return $\pi^{[R]}$.

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 [18]) 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 $\mathcal{Y} = \{(s_1,a_1,y_1),\dots,(s_n,a_n,y_n)\}$. Define the design matrix, the noise vector, and the response vector

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

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 ξ, and the second is the deterministic ridge bias toward zero induced by λ. 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. [1] for linear regression with conditionally σ_r-sub-Gaussian noise, with probability at least $1-\delta$, uniformly over all labeled sets $\mathcal{Y} \subseteq \mathcal{D}$ the learner may encounter,

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

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_{π} is the one-step feature transition operator under π, 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. [18] adapted to retained-reward partial labels by Hu et al. [15]—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. [1]. By construction, β_{off}, σ_{off}, and $\Lambda(\mathcal{D})$ are functions of $(\mathcal{D},\delta)$ only.

Theorem 4.4 of Jin et al. [18], instantiated in the linear-MDP setting of Definition 1 and combined with the pessimistic-reward clipping of Hu et al. [15], 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. [18], Hu et al. [15]; 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 π. 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 μ 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 Λ 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 Δ, $\Phi(\mathcal{Y}^{[b]} \cup \{(s, a)\}; \mu) = \Phi(\mathcal{Y}^{[b]}; \mu) - \Delta(s, a; \mathcal{Y}^{[b]}, \mu) \leq \Phi(\mathcal{Y}^{[b]}; \mu)$, so Φ 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 μ, 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 μ, 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 μ 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 Δ 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 ν 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 ν 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 ν 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 μ, 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 κ, $\Sigma(d^{\pi^\star_r}) \preceq c_{\mathrm{beh}}\, \Sigma(\hat{d}^{(b)}_{\mathrm{beh}})$. 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 ν and any labeled set $\mathcal{Y}$ producing a pessimistic policy $\hat{\pi}_{\tilde{r}}$ with confidence radius α,*

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

Proof. Theorem 1, applied to $\mathcal{Y}$ and the pessimistic policy $\hat{\pi}_{\tilde{r}}$ with confidence radius α, 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 σ 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 ν 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 ν 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 $x^\top Mx = \mathrm{tr}(Mxx^\top)$. 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 ν,*

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

Proof. Theorem 2 applied at the terminal round B with reference measure ν 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 ρ_b.

For a fixed round b, reference measure μ, 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 μ, 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 Σ, $\|\Sigma^{1/2}v\|_2^2 = v^\top \Sigma v$. 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 ζ_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 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 Λ_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. □

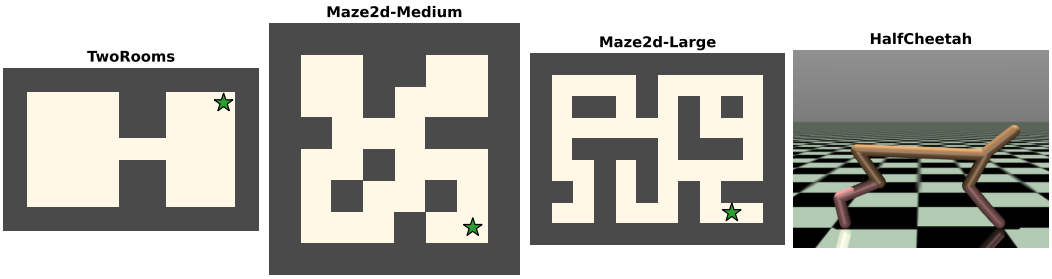

Figure 4: Visualization of the four evaluation domains. From left to right: TwoRooms, Maze2d-Medium, Maze2d-Large, and HalfCheetah-Medium-Replay.

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 5.4) 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 five domains: the finite linear MDPs of Section 5.1, constructed so that the assumptions of Section 3 hold exactly, and 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 [12]. 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 [5, 12]. 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. [7]. 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. [7], 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. [26], 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 [24] 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 Λ_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 [20] hyperparameters are listed in Table 4; we follow standard IQL settings throughout.

Table 4: IQL hyperparameters.

Hyperparameter	Value
Hidden sizes for V, Q, π	256×256 MLP, ReLU
Optimizer	Adam
Learning rate	3×10^{-4} (all three networks)
Batch size	256 (D4RL); 64 (TwoRooms)
Discount γ	0.99 (HC-MR); 0.995 (Maze2d-M, Maze2d-L); 0.95 (TwoRooms)
Target-network Polyak rate τ_{soft}	5×10^{-3}
Expectile parameter τ	0.7 (D4RL); 0.5 (TwoRooms)
Advantage-weighting temperature β	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 5. 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 5: 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 β_b	0.1 (constant across rounds)
Successor-feature ridge regularizer λ_{SF}	10^{-3}
Successor-feature estimator	closed-form ridge (Appendix B.2)
Warm-start labels B_0	64, uniformly sampled; shared across methods and counted toward the budget
Acquisition batch size M	64
Refit cadence	every acquisition round (after each batch of M labels)

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.

E.6 Computational Cost

The methods differ in the computation required for acquisition. UNIFORM selects labels without fitting an acquisition model and is therefore substantially cheaper. UNCERTAINTY updates only the reward estimator and its uncertainty quantities, whereas RLLF-GUIDED, REACH, IDRL, and SURE additionally update policy-, value-, successor-, or policy-evaluation-related quantities. SURE's method-specific overhead consists of recomputing successor features, constructing the high-value target set, evaluating reachability, and scoring the unlabeled candidates. As described in Appendix A, these components are updated once per batch of $M = 64$ newly acquired labels rather than after every individual label.

We compare the wall-clock time of all methods under identical batch sizes, feedback budgets, hardware, and evaluation protocols. Across the evaluated domains, SURE takes on average approximately $1.05\times$ as long as RLLF-GUIDED and IDRL, and has comparable runtime to REACH and UNCERTAINTY. The successor-feature and reachability computations used by SURE and REACH, as well as the reward-uncertainty updates used by SURE and UNCERTAINTY, contribute only a small fraction of the total runtime relative to the repeated offline-learning updates. UNIFORM, which does not require adaptive model-based acquisition, remains the least computationally expensive method overall.

Relative to UNIFORM, the results expose an explicit label–compute trade-off. For example, on Maze2d-Large, SURE uses approximately $20\times$ more computation but reaches 90% of the matched full-feedback return with approximately $30\times$ fewer reward labels. Thus, SURE is more computationally expensive than simple non-adaptive selection, but its wall-clock cost is comparable to existing informed adaptive approaches while yielding substantially greater label efficiency.

E.7 Sensitivity to the Warm-Start Size

Early in acquisition, the reward model and the reachability computed from its policy and value estimates may be unreliable. In all continuous-control experiments, adaptive acquisition therefore begins from $B_0 = 64$ uniformly sampled reward labels (Appendix A). To evaluate sensitivity to this choice, we vary the warm-start size $B_0 \in \{64, 200, 400\}$ while keeping the subsequent acquisition batch size fixed at $M = 64$. Table 6 reports the percentage feedback required to reach 80% of the matched full-feedback return, averaged over 50 paired seeds.

Table 6: Percentage feedback required by SURE to reach 80% of the matched full-feedback return under different warm-start sizes B_0, averaged over 50 paired seeds.

Domain	$B_0 = 64$	$B_0 = 200$	$B_0 = 400$
Maze2d-Large	**0.060%**	0.064%	0.067%
HalfCheetah	**0.300%**	0.330%	0.360%

The feedback required by SURE changes only marginally across the tested warm-start sizes, showing that its label efficiency is stable with respect to the initialization. This robustness is consistent with the design of the SURE score: the reachability measure (10) mixes the successor-guided distribution with the empirical behavior distribution through β_b, so inaccurate early value estimates cannot fully determine the acquisition distribution, and the multiplicative uncertainty factor in (11) further requires selected transitions to be informative for the reward model. The mild advantage of $B_0 = 64$ is also expected: fewer labels are spent on uniform initialization, allowing adaptive acquisition to begin earlier.

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. Finally, the offline-RL backbones considered in this work output continuous actions only; extending retained-label reward selection to more challenging settings, such as parameterized action spaces that combine discrete and continuous components, could be a direction for future work [41, 42, 44, 43].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the four main contributions—formulating retained-label reward selection, deriving a label-sensitive performance bound, introducing the SURE acquisition rule, and analyzing it through coverage transfer and budget-dependent rates—and each is supported in the corresponding sections (Sections 3–5) and appendices.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Every theorem, proposition, and corollary is stated with explicit assumptions and has a complete proof in the appendix: Appendix C for the Section 4 results and Appendix D for coverage transfer, the unified bound, and the budget-dependent rates.

   Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix E describes the five domains (with versions of the public D4RL datasets), the implementation of every retained-label reward-selection baseline, the partially labeled offline-RL pipeline, and IQL/SURE hyperparameters; the SURE algorithm pseudocode is in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments use the publicly available D4RL datasets; we also submitted our code in the supplementary file.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 reports the experimental setting (five domains, six strategies, fifty seeds, full-feedback reference); Appendix E reports dataset versions, baseline implementations, IQL hyperparameters (Table 4), SURE hyperparameters (Table 5), and the evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Each curve in Figure 2 is the mean over 50 random seeds, with shaded regions showing the standard error of the mean; this convention is stated in Section 5 and in the Evaluation Protocol of Appendix E.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we provided in the Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics; it does not involve human subjects, sensitive personal data, or systems with foreseeable misuse risks.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is foundational research on offline reward selection for reinforcement learning under limited feedback, with no direct deployment path or specific application that would entail foreseeable societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not have pre-trained models, generative systems, or scraped datasets that pose dual-use risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use the publicly released D4RL datasets [12] (Apache 2.0) and cite the original sources for each algorithmic baseline—PEVI [18], PDS [15], IQL [20], RLLF [6], IDRL [26], and FQE [24]—as well as the OpenAI Gym MuJoCo environments [5].

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not have new datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects, so no IRB approval is required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the core methodology, scientific rigor, or originality of this research; any LLM usage was limited to writing and does not affect the results.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.