Seeking Disagreement: Online Credit Assignment with Delayed and Pseudo-Aggregated Rewards

(One-Page Extended Abstract)

Many reinforcement learning problems provide only sparse or delayed feedback: an agent makes a sequence of primitive decisions before observing a single outcome, such as an aggregate sales report, a final clinical test result, or a goal-reaching indicator. This creates a credit-assignment problem because the learner observes the trajectory of states and actions but not the primitive rewards associated with individual decisions. Standard model-free methods can in principle propagate the delayed outcome backward through temporal-difference updates or value-function bootstrapping, but this propagation can be slow and unstable when the delay horizon is long and successful outcomes are rare. Alternatively, a direct block-level formulation avoids primitive reward recovery by treating an entire length-T action sequence as one macro-action, but this induces an action space whose size grows exponentially in T. This project studies a third approach: learn a primitive pseudo-reward from delayed aggregate or endpoint reward, and then use that recovered primitive reward for online policy learning.

The proposed framework combines two groups of reward models: The first group consists of flexible reference reward models. These are designed to be relatively low-bias, data-driven and consistent, but can be high variance when delayed labels are scarce. In the tabular experiments, the reference model is ridge regression on state-action visitation counts; in the continuous PointMaze experiment, it is a deep neural reward model trained from trajectory-level labels. The second group consists of structured surrogate reward models. These models encode lower-dimensional domain knowledge, such as linear economic structure, clinical health regimes, spatial maze regions, or geometric navigation heuristics. Surrogates can provide useful low-variance signals early in training, but they may be misspecified. The algorithm estimates local reliability through MSE or uncertainty proxies, forms an adaptive mixed reward, and uses disagreement between the reference estimator and the locally credible surrogate as an optimistic exploration signal.

A central contribution is that the framework is useful not only when the observed feedback is a true additive aggregate, but also when it is only an endpoint outcome. In the Toy Advertiser MDP, the observed block return is a noisy sum of primitive profit, so reward recovery is well-specified. In the Sepsis simulator, I consider both additive feedback and terminal-outcome feedback; in the terminal case, the delayed observable health outcome is not literally the sum of primitive rewards, but a pseudo-additive reward estimation can still serve as a credit-assignment heuristic. In PointMaze, the entire trajectory receives a sparse binary endpoint label, and the learned primitive reward is again a pseudo-reward used to guide SAC. Thus, the project tests both reward recovery under true delayed aggregation and pseudo-reward learning under sparse outcome supervision.

I evaluate the method in three environments. The first is a tabular Toy Advertiser MDP, where states represent shopper population and actions represent advertising intensity. The baselines are Delayed Q-learning, Block-UCB, Ridge-UCB, Adaptive MSE-UCB, and Disagreement-UCB [1]. This experiment isolates the tabular mechanism: Adaptive MSE-UCB and Disagreement-UCB use the same candidate models and adaptive reward mixture, but differ in the exploration bonus. The results show that reward-recovery methods substantially outperform delayed Q-learning and Block-UCB. Adaptive MSE-UCB and Disagreement-UCB achieve the lowest regret, while Disagreement-UCB obtains the lowest policy error and mixed reward MSE. Thus, the clean tabular experiment supports the main hypothesis that surrogate-assisted reward recovery improves learning and that disagreement is useful for recovering reward structure.

The second experiment uses a structured Sepsis treatment simulator, which is a larger episodic MDP. States include diabetes status, vital signs, laboratory abnormality, and active interventions; actions are the eight combinations of antibiotics, vasopressors, and ventilation. The methods are Random, Delayed DQN, Ridge pseudo-reward UCB, Adaptive MSE-UCB, and Disagreement-UCB. Surrogates include a linear clinical heuristic and a health-regime table. This experiment separates two cases: additive feedback, where the observed label is a true cumulative clinical utility, and terminal feedback, where the label is only the final health outcome. The latter case is conceptually important because the recovered reward should be interpreted as a pseudo-reward rather than a true clinical primitive reward.

The third experiment extends the framework to continuous control using PointMaze Medium. The simulator produces a discrete-time sequence of continuous states and actions, and we treat the entire trajectory of maximum length H as one delayed-feedback sample. Each trajectory receives a single binary endpoint label indicating whether the agent reaches the goal within H steps. SAC is used as the policy optimizer. The methods are Delayed SAC, DNN-reference SAC, Adaptive MSE-SAC, and Disagreement-SAC. The reward models are a DNN reference model, a box-region surrogate, and a heuristic-feature neural surrogate. Since tabular count-based UCB is not meaningful in continuous state-action spaces, SAC entropy provides generic exploration, while MSE and disagreement provide reward-space optimism. The resulting success-rate curves are noisy, but they show that learned pseudo-rewards can improve continuous-control learning from sparse trajectory labels.

Overall, the experiments support the hypothesis that structured surrogate models can accelerate learning from delayed sparse feedback when combined with a more flexible reference estimator to reduce the risk of misspecification. Disagreement provides both a diagnostic of local model conflict and a targeted exploration signal.

[1] Details of the baseline algorithms are provided in Appendix A.

Seeking Disagreement: Online Credit Assignment with Delayed and Pseudo-Aggregated Rewards

Haozhan Gao* Jason Meng† Jose Blanchet†

June 9, 2026

Abstract

Reward signals in reinforcement learning are often sparse, delayed, or observed only as aggregate outcomes over many primitive decisions. This makes online credit assignment and policy learning difficult: the learner sees the trajectory but does not know which state-action pairs caused the observed outcome. I study a surrogate-assisted reward recovery framework that fits primitive pseudo-reward models from delayed labels. The algorithm combines a flexible reference estimator with structured surrogate reward models, adaptively mixes them using local MSE or uncertainty proxies, and uses disagreement between the reference estimator and the locally credible surrogate as exploration signal for model selection. I evaluate the idea in three settings: a tabular Toy Advertiser MDP with truly aggregated block rewards, a structured Sepsis simulator with additive and terminal-outcome feedback, and a continuous PointMaze robot task with trajectory-level sparse endpoint labels. The experiments show that reward recovery improves over several direct delayed-feedback baselines including the commonly used model-free methods. They also illustrate a broader contribution: even when the observed outcome is not literally a sum of primitive rewards, a learned pseudo-reward can still provide useful credit assignment for policy learning.

1 Introduction

Delayed and aggregated feedback appears in many sequential decision problems. In advertising, the effect of daily promotion decisions may only be visible in weekly or monthly sales. In healthcare, a sequence of treatment decisions may be evaluated by a final clinical outcome. In robot navigation, a long trajectory may receive only a binary success label. These settings are challenging because the learner observes the trajectory of states and actions but not the primitive reward at each time step.

A direct model-free baseline can place the observed outcome at the end of the trajectory and run Q-learning, DQN, SAC, or PPO. This is simple, but the learning signal is sparse and delayed. Another baseline treats an entire length-T action sequence as a macro-action and plans at the block level. This avoids primitive reward recovery but creates a composite action space of size $|\mathcal{A}|^T$. We instead recover a primitive pseudo-reward from delayed labels and use it for primitive-time policy learning.

The core difficulty is a bias-variance tradeoff. A flexible data-driven reward estimator can be robust to misspecification, but under delayed feedback it is high variance. A structured surrogate can generalize quickly, but may be biased. We propose to fit both classes of estimators, estimate their local reliability, mix them adaptively, and explore where a locally credible surrogate disagrees with the reference estimator. This disagreement is useful because it identifies regions where additional data can distinguish between competing credit-assignment explanations.

Our contributions are threefold. First, we formulate a disagreement-seeking reward recovery algorithm for online RL with delayed aggregate feedback. Second, we extend the idea to multiple surrogates and to pseudo-reward learning when the observed label is a terminal or trajectory-level outcome rather than a literal sum of primitive rewards. Third, we evaluate the method in three environments of increasing complexity: a tabular advertiser MDP, a structured Sepsis simulator, and continuous PointMaze with SAC.

2 Related Work

Delayed and aggregated feedback. Delayed feedback has been studied in online learning and bandits, including general reductions (Joulani et al., 2013), delayed conversions (Vernade et al., 2017), and delayed aggregated anonymous feedback (Pike-Burke et al., 2018; Cesa-Bianchi et al., 2019). In reinforcement learning, temporal-difference learning and eligibility traces propagate rewards backward through bootstrapping (Sutton, 1988; Singh and Sutton, 1996), but they still require

*Corresponding author for the CS224R final project. Stanford Graduate School of Business. Email: haozhan.gao@stanford.edu.
†Outside collaborators. Department of Management Science and Engineering, Stanford University.

observed rewards. Our setting is more severe: the learner receives only block-level or trajectory-level labels and must infer a useful primitive reward model.

Reward shaping and surrogate rewards. Potential-based shaping can preserve the optimal policy under specific reward transformations (Ng et al., 1999; Wiewiora, 2003; Devlin and Kudenko, 2012). Other work studies shaping and intrinsic rewards to accelerate sparse-reward learning (Mataric, 1994; Randløv and Alstrøm, 1998; Grzes and Kudenko, 2009; Kulkarni et al., 2016). In contrast, our surrogate rewards are not assumed to be policy-invariant. They are fitted from the same delayed labels as the reference estimator and are treated as low-dimensional, potentially biased reward hypotheses.

Uncertainty and disagreement. Exploration based on uncertainty appears in UCB-style RL (Auer, 2002; Auer et al., 2008; Strehl and Littman, 2008; Azar et al., 2017; Jin et al., 2018), bootstrapped or randomized value functions (Osband et al., 2016, 2018), and ensemble uncertainty (Lakshminarayanan et al., 2017). Intrinsic motivation and model disagreement have also been used to drive exploration (Houthooft et al., 2016; Burda et al., 2018; Chua et al., 2018; Pathak et al., 2019; Sekar et al., 2020). Our disagreement signal is targeted to reward recovery: it compares a flexible reference reward estimator with structured surrogate reward estimators.

Sepsis and off-policy evaluation. Our Sepsis experiment is inspired by the synthetic treatment environment used in counterfactual off-policy evaluation (Oberst and Sontag, 2019). We use it not for off-policy evaluation, but as a structured environment where terminal outcomes and clinical reward heuristics create a natural delayed credit-assignment problem.

3 Problem Formulation

3.1 Delayed aggregate feedback and pseudo-aggregate feedback

We consider a primitive-time MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r)$ in which the learner observes states, actions, and transitions at every primitive time step, but does not directly observe the primitive reward. In the ideal delayed-aggregate setting, time is divided into blocks of length T. For block i, the learner observes

$$y_i = \sum_{t=(i-1)T}^{iT-1} r(s_t, a_t) + \epsilon_i, \tag{1}$$

where ϵ_i is block-level noise. The learning objective is not to choose a block-level action sequence, but to recover a useful primitive-time reward signal and use it to learn a primitive-time policy. This distinction is important because a block-MDP formulation has action space $|\mathcal{A}|^T$, whereas reward recovery keeps planning and control at the original time scale.

A central point of this project is that the same machinery can also be useful when the observed label is not literally generated by (1). In the Sepsis terminal-feedback experiment, the label is the final health outcome. In PointMaze, the label is a binary trajectory success indicator. In these cases, there may not exist a true primitive reward r whose sum exactly equals the observed label. We nevertheless fit an additive primitive model

$$y_i \approx \sum_{t \in i} \tilde{r}(s_t, a_t) \tag{2}$$

and interpret $\tilde{r}$ as a *pseudo-reward*. The pseudo-reward is not claimed to be the ground-truth reward decomposition; it is a learned credit-assignment signal used for exploration and policy improvement. This is one of the main conceptual extensions tested in the experiments.

3.2 Reference reward models: consistent and low-bias estimators

The first group of reward estimators is a flexible reference model. Its role is to provide a relatively low-bias, data-driven consistent estimate of primitive reward or pseudo-reward. In the tabular experiments, the reference model is ridge pseudo-reward recovery with a small regularization hyperparameter. For block i, let $x_i \in \mathbb{R}^{SA}$ be the state-action count vector, where $x_i(s, a)$ is the number of times pair (s, a) appears in the block. Stacking the observed blocks gives

$$y_k = X_k \theta + \varepsilon \tag{3}$$

where X_k is a counting matrix and $\theta(s, a) = r(s, a)$ is a vector of primitive rewards. The ridge reference estimator is

$$\tilde{\theta}_k = (X_k^\top X_k + \lambda I)^{-1} X_k^\top y_k, \qquad \tilde{r}_k(s, a) = \tilde{\theta}_k(s, a). \tag{4}$$

This estimator is model-agnostic: it does not impose a parametric structure on the reward table beyond tabularity. Its advantage is robustness to surrogate misspecification, and in the well-specified additive setting it targets the primitive reward table directly. Its disadvantage is high variance and slow learning when few aggregate labels have been observed.

In the continuous PointMaze experiment, an explicit tabular ridge table is no longer meaningful. The reference model is therefore a DNN reward model $\tilde{r}_\psi(s,a)$ with input $\big((x,y,v_x,v_y),(a_x,a_y)\big)$. It is trained from trajectory-level labels by minimizing

$$\sum_i \left(y_i - \sum_{t=0}^{L_i-1} \tilde{r}_\psi(s_{i,t},a_{i,t}) \right)^2 + \lambda_{\text{sparse}} \sum_{i,t} |\tilde{r}_\psi(s_{i,t},a_{i,t})|. \tag{5}$$

This DNN plays the same algorithmic role as ridge: it is the flexible reference estimator against which lower-complexity surrogates are compared. We do not require it to be an perfectly unbiased estimator in finite samples; rather, it serves as the least structurally constrained reward model in the candidate family.

3.3 Surrogate reward models: structured, low-variance estimators

The second group consists of surrogate reward models. Each surrogate model r_m encodes a simple structural hypothesis about the reward function, often motivated by domain knowledge about the environment or decision context. It is fitted from exactly the same delayed labels as the reference model, using the additive trajectory-level objective

$$\min_{\theta_m} \sum_i \left(y_i - \sum_{t\in i} \hat{r}_m(s_t,a_t;\theta_m) \right)^2 + \lambda_m \|\theta_m\|_2^2. \tag{6}$$

Surrogates can be substantially lower variance because they pool information across many primitive state-action pairs. However, they have the inductive bias if their structural assumptions are wrong. This bias-variance tradeoff is the motivation for combining them with the reference model rather than using them alone.

The surrogate family changes by experiment. In the Toy Advertiser MDP, the linear surrogate uses features $(1,s,a,sa)$ and the threshold surrogate learns a separate action table for low states $s \le 4$ and high states $s \ge 5$. In Sepsis, the linear surrogate uses abnormality count, current treatment burden, action treatment intensity, and diabetes status; the health surrogate is a 3×3 table over patient regime and action intensity. In PointMaze, the box surrogate assigns one scalar reward to each free 1×1 maze cell, while the heuristic neural surrogate uses three domain features: Euclidean distance to the goal, number of wall cells intersecting the straight line to the goal, and distance to the next wall if the current action is maintained.

Thus, the algorithm always compares two groups of estimators: a flexible reference estimator, which is intended to reduce asymptotic bias, and one or more structured surrogate estimators, which provide low-variance inductive bias. The full method uses local MSE proxies to determine how much weight to assign to each model, and uses reference-surrogate disagreement as an exploration signal. Intuitively, when data are scarce, the structured surrogates may outperform the flexible reference estimator because they pool information through stronger inductive assumptions and therefore have lower variance. As more data are collected, however, the reference estimator can use its greater flexibility to produce more accurate local reward estimates, so the adaptive weighting scheme gradually shifts more weight toward the reference model.

4 Method

4.1 Disagreement, bias, and local MSE proxies

Let the candidate model set be

$$\mathcal{M}^+ = \{\text{ref}\} \cup \mathcal{M},$$

where the reference model is the tabular ridge estimator in Experiments 1–2 and the DNN reference model in Experiment 3. For each model $j \in \mathcal{M}^+$, let $\hat{r}_{j,k}(s,a)$ denote its fitted primitive reward prediction at round k. A key diagnostic is the local disagreement between the reference estimator and a surrogate model. For a surrogate $m \in \mathcal{M}$, define the squared disagreement

$$\Delta^2_{m,k}(s,a) = \left(\hat{r}_{\text{ref},k}(s,a) - \hat{r}_{m,k}(s,a)\right)^2. \tag{7}$$

This quantity is informative because it can be decomposed into variance and bias terms. Suppressing the dependence on (s,a), we have

$$\mathbb{E}\left[\left(\hat{r}_{\text{ref},k} - \hat{r}_{m,k}\right)^2\right] = \text{MSE}(\hat{r}_{\text{ref},k}) + \text{MSE}(\hat{r}_{m,k}) - 2\,\text{Cov}(\hat{r}_{\text{ref},k},\hat{r}_{m,k}) + \text{bias-interaction terms}. \tag{8}$$

More concretely, if the reference estimator is treated as approximately unbiased relative to the surrogate, then bias-interaction terms can be ignored, and a large reference-surrogate gap indicates either high estimator uncertainty, surrogate misspecification, or both. Thus disagreement is useful in two ways: it contributes to a surrogate's local MSE proxy, and it also identifies regions where additional samples are useful for distinguishing between the flexible reference model and the structured surrogate.

In the tabular experiments, we estimate this quantity by trajectory-level bootstrap. For bootstrap draw $b = 1, \ldots, B$, we resample trajectories with replacement, refit every reward model, and obtain bootstrap predictions $\hat{r}_{j,k}^{(b)}(s,a)$. The bootstrap disagreement proxy is

$$\widehat{\Delta}_{m,k}^2(s,a) = \frac{1}{B}\sum_{b=1}^{B}\left(\hat{r}_{\mathrm{ref},k}^{(b)}(s,a) - \hat{r}_{m,k}^{(b)}(s,a)\right)^2. \tag{9}$$

The reference model's local MSE proxy is its bootstrap variance:

$$\widehat{\mathrm{MSE}}_{\mathrm{ref},k}(s,a) = \widehat{\mathrm{Var}}(\hat{r}_{\mathrm{ref},k}(s,a)). \tag{10}$$

For each surrogate m, we use the proxy

$$\widehat{\mathrm{MSE}}_{m,k}(s,a) = \widehat{\mathrm{Var}}(\hat{r}_{m,k}(s,a)) + \widehat{\Delta}_{m,k}^2(s,a). \tag{11}$$

This expression reflects the bias–variance role of surrogate models. The variance term captures estimation uncertainty in the surrogate, while the disagreement term captures possible misspecification relative to the flexible reference estimator. This proxy is not intended to be a calibrated confidence interval; it is a local reliability score used for reward mixing and exploration.

In the continuous PointMaze experiment, refitting many bootstrap DNNs is computationally expensive. We therefore use cheaper local uncertainty proxies: MC-dropout variance for neural reward models and analytic ridge variance for the box-region surrogate. The same principle is retained: the reference model uses its own uncertainty proxy, while surrogate MSE proxies add uncertainty to squared disagreement with the reference model.

4.2 Adaptive reward mixing

Given the local MSE proxies $\widehat{\mathrm{MSE}}_{j,k}(s,a)$, the algorithm forms an adaptive mixture of the candidate reward models. The goal is not to choose a single global model, but to trust different models in different regions of the state-action space. A surrogate may be preferable in regions where data are scarce and its structural assumptions reduce variance, while the reference estimator may become preferable in regions where enough data have been collected and flexibility matters more than inductive bias.

We convert MSE proxies into relative-MSE softmax weights. For each (s,a), define

$$m_{\min,k}(s,a) = \min_{j\in\mathcal{M}^+}\widehat{\mathrm{MSE}}_{j,k}(s,a), \qquad \mathrm{scale}_k(s,a) = \max\left\{\mathrm{median}_{j\in\mathcal{M}^+}\widehat{\mathrm{MSE}}_{j,k}(s,a), \epsilon\right\}. \tag{12}$$

Then define the relative excess MSE

$$\delta_{j,k}(s,a) = \frac{\widehat{\mathrm{MSE}}_{j,k}(s,a) - m_{\min,k}(s,a)}{\mathrm{scale}_k(s,a)}. \tag{13}$$

The adaptive model weight is

$$w_{j,k}(s,a) = \frac{\exp\left(-\delta_{j,k}(s,a)/\alpha\right)}{\sum_{\ell\in\mathcal{M}^+}\exp\left(-\delta_{\ell,k}(s,a)/\alpha\right)}. \tag{14}$$

Here $\alpha > 0$ controls how sharply the mixture concentrates on the lowest-MSE model. Smaller α approaches hard local model selection, while larger α produces smoother averaging.

The mixed reward used for planning or policy optimization is

$$\bar{r}_k(s,a) = \sum_{j\in\mathcal{M}^+} w_{j,k}(s,a)\hat{r}_{j,k}(s,a). \tag{15}$$

This construction implements the intended bias–variance tradeoff. Early in learning, structured surrogates often receive larger weights because they pool information across states and actions and therefore have lower variance. As more data are collected, the flexible reference estimator can produce more accurate local reward estimates, and the weights can shift toward the reference model. In this sense, the mixture uses surrogate structure for sample efficiency without permanently committing to a potentially misspecified reward model.

4.3 Disagreement-seeking reward optimism

Let

$$j_k^*(s,a) \in \arg\min_j \widehat{\mathrm{MSE}}_{j,k}(s,a) \tag{16}$$

be the locally best model. Adaptive MSE-UCB uses the reward

$$R_k^{\mathrm{MSE}}(s,a) = \bar{r}_k(s,a) + b_{\mathrm{count},k}(s,a) + c_{\mathrm{mse}} \frac{\sqrt{\widehat{\mathrm{MSE}}_{j_k^*,k}(s,a)}}{\sqrt{\lambda + N_k(s,a)}}. \tag{17}$$

Disagreement-UCB uses the same mixed reward and count bonus, but changes the model-dependent exploration signal. If the locally best model is the reference estimator, it falls back to reference uncertainty; if the locally best model is a surrogate, it uses reference-surrogate disagreement:

$$e_k(s,a) = \begin{cases} \sqrt{\widehat{\mathrm{MSE}}_{\mathrm{ref},k}(s,a)}, & j_k^*(s,a) = \mathrm{ref}, \\ \sqrt{\widehat{\Delta}^2_{j_k^*,k}(s,a)}, & j_k^*(s,a) \in \mathcal{M}. \end{cases} \tag{18}$$

The planning reward is

$$R_k^{\mathrm{Dis}}(s,a) = \bar{r}_k(s,a) + b_{\mathrm{count},k}(s,a) + c_{\mathrm{dis}} \frac{e_k(s,a)}{\sqrt{\lambda + N_k(s,a)}}. \tag{19}$$

The tabular agents then estimate $\hat{P}_k$ from transition counts and run value iteration with R_k to obtain the data-collection policy.

4.4 Continuous extension with SAC

In the tabular experiments, optimistic planning uses value iteration with a count-based bonus of the form $1/\sqrt{N(s,a)}$. This construction does not directly extend to continuous state-action spaces, where exact visit counts are unavailable and tabular dynamic programming is infeasible. In the continuous PointMaze experiment, we therefore use SAC as the policy optimizer. SAC provides exploration through its entropy-regularized objective, so we do not introduce an additional tabular count-based UCB term.

At round k, the reward-recovery module produces a primitive reward estimate $\hat{r}_k(s,a)$, which may be a reference reward estimate, an adaptive mixture, or an optimistic disagreement-based reward. We use this estimated reward to relabel the primitive transitions stored in the replay buffer:

$$\tilde{r}_{i,t}^{(k)} = \hat{r}_k(s_{i,t}, a_{i,t}).$$

SAC is then trained on the relabeled transition dataset

$$(s_{i,t}, a_{i,t}, \tilde{r}_{i,t}^{(k)}, s_{i,t+1}),$$

using its standard soft objective. After training, the resulting stochastic SAC actor is used to collect new online trajectories, whose trajectory-level labels are appended to the dataset for the next reward-model refit. Evaluation is performed separately using the deterministic mean action of the actor; these evaluation rollouts are used only for measuring success rate and are not added to the replay buffer.

The full disagreement-seeking algorithms for tabular and continuous MDP settings are provided in Appendix B.

5 Theory: Average-Reward Regret for the Disagreement-Seeking Component

The full algorithm includes adaptive weighting, multiple surrogates, and continuous-control variants. The theory below analyzes the tabular disagreement-seeking component under the ideal delayed-aggregate model. This theory is not intended to fully characterize the continuous SAC implementation; rather, it explains why primitive reward recovery can avoid the exponential cost of block-level planning and why the regret depends only as $\sqrt{T}$ on the delay horizon under occupancy regularity.

We analyze a communicating tabular MDP over N primitive steps. Rewards are unobserved at primitive time, and the learner observes block labels

$$y_m = \sum_{t=1}^{T} r(s_{mT+t}, a_{mT+t}) + \epsilon_m, \tag{20}$$

where ϵ_m is σ-subGaussian. Regret is measured at the primitive time scale:

$$Reg(N) = \sum_{k=1}^{N} (g^* - r(s_k, a_k)). \tag{21}$$

Assumption 1 (Occupancy Regularity). *There exists $\kappa \geq 1$ such that each completed block count vector x_m satisfies $\|x_m\|_2^2 \leq \kappa T$.*

Assumption 2 (Uniformly Bounded Disagreement). *There exists $\Delta_{\max} \geq 0$ such that for all rounds and all (s, a), the disagreement signal is at most $\Delta_{\max}$.*

Assumption 3 (Diagonal Dominance). *There exists $\eta \in (0, 1]$ such that the ridge Gram matrix $V_e = X_e^\top X_e + \lambda I$ satisfies $V_e \succeq \eta \operatorname{diag}(V_e)$ at every epoch.*

Theorem 1 (Average-reward regret bound). *Assume $r(s, a) \in [0, 1]$, the MDP is communicating with diameter D, block noise is σ-subGaussian, and Assumptions 1–3 hold. Then there exists a choice of optimistic planning radii such that, for any $N \geq 1$, with probability at least $1 - \delta$,*

$$Reg(N) \leq \tilde{O}\left(\left(D + \sigma\sqrt{\frac{\kappa T}{\eta}} + \Delta_{\max}\right)\sqrt{SAN}\right), \tag{22}$$

where $\tilde{O}(\cdot)$ hides logarithmic factors in $(SA, N, 1/\delta, \lambda)$.

Proof. See Appendix C.2. □

6 Experimental Setup

The experiments are designed to separate three questions. First, when the observed label is truly additive, does primitive reward recovery improve over delayed model-free learning and block-level planning? Second, when the label is not truly additive, can a pseudo-reward still improve policy learning? Third, holding the reward model family fixed, does disagreement-based optimism improve over an uncertainty-only adaptive mixture?

6.1 Experiment 1: Toy Advertiser MDP

Experiment 1 is a small tabular continuing MDP designed to isolate the delayed-aggregate reward recovery mechanism. The state $s \in \{0, \ldots, 9\}$ represents a shopper population index, and the action $a \in \{0, 1, 2\}$ represents low, medium, or high advertising intensity. From an interior state, action 0 moves the population down with probability $p_\downarrow = 0.6$, action 2 moves it up with probability $p_\uparrow = 0.6$, and action 1 moves up or down with probabilities $p_\uparrow^{\text{mid}} = p_\downarrow^{\text{mid}} = 0.3$. Boundary states reflect.

The true primitive reward is

$$r(s, a) = 0.05s - 0.005s^2 - 0.05a - 0.12a^2 + \phi(a)s, \qquad \phi(0) = 0,\ \phi(1) = 0.02,\ \phi(2) = 0.05. \tag{23}$$

The quadratic term in s creates diminishing returns from high population, the quadratic term in a creates convex advertising cost, and the interaction term captures the fact that advertising is more valuable when there are more potential customers. The learner observes only noisy block returns

$$Y_i = \sum_{t \in i} r(s_t, a_t) + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, 0.2^2), \tag{24}$$

with block length $T = 7$.

We compare five methods. Delayed Q-learning is the model-free delayed-feedback baseline: it assigns zero reward to nonfinal positions in a block and assigns Y_i to the last transition. Block-UCB is the direct block-MDP baseline: it treats a length-T primitive action sequence as one macro-action. Ridge-UCB is the model-agnostic reward recovery baseline: it uses only the ridge pseudo-reward table. Adaptive MSE-UCB and Disagreement-UCB use the same model family $\{\text{ridge}, \text{linear}, \text{threshold}\}$ and the same relative-MSE reward mixture. The only difference is the optimism bonus: Adaptive MSE-UCB uses the MSE of the locally best model, while Disagreement-UCB uses reference-surrogate disagreement whenever a surrogate is locally best. Full algorithmic details are in Appendix A.1.

6.2 Experiment 2: Sepsis simulator

Experiment 2 tests whether the method remains useful in a larger structured episodic MDP with clinically interpretable surrogate models. The state is

$$s = (\text{diabetes}, \text{heart rate}, \text{blood pressure}, \text{oxygen}, \text{glucose}, \text{antibiotics on}, \text{vasopressors on}, \text{ventilator on}). \tag{25}$$

The four vitals/labs take three discrete levels, the three treatments are binary, and diabetes is binary, giving 1296 states. An action is a binary treatment vector $(a_{\text{abx}}, a_{\text{vaso}}, a_{\text{vent}}) \in \{0,1\}^3$, so there are 8 actions. Transitions are generated by the simulator: antibiotics primarily affect heart rate, vasopressors affect blood pressure, ventilator affects oxygen, and glucose dynamics depend on diabetes and treatment intensity.

Let $A(s)$ denote the number of abnormal vitals/labs and $U(s)$ denote the number of active treatments. The true simulator reward is evaluated after the transition:

$$r_{\text{true}}(s') = \begin{cases} +1, & A(s') = 0 \text{ and } U(s') = 0, \\ -1, & A(s') \geq 3, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

The horizon is $T = 5$. We run two feedback modes. In additive mode, the observed label is $Y_i = \sum_{t=0}^{T-1} r_{\text{true}}(s_{i,t+1})$, which matches the delayed aggregate reward assumption. In terminal mode, the observed label is $Y_i = r_{\text{true}}(s_{i,T})$. The terminal mode is intentionally misspecified for additive reward recovery; success there would indicate that pseudo-reward learning can still help policy learning even when the label is an endpoint outcome rather than a sum of primitive rewards.

The methods are Random, Delayed DQN, Ridge pseudo-reward UCB, Adaptive MSE-UCB, and Disagreement-UCB. Random is a lower bound. Delayed DQN is the model-free delayed-label baseline, assigning the observed window label only to the final step. Ridge pseudo-reward UCB is the flexible reference-only baseline. Adaptive MSE-UCB and Disagreement-UCB use the model family $\{\text{ridge}, \text{linear}, \text{health}\}$. The linear surrogate uses features based on abnormality count, current treatment count, action treatment intensity, and diabetes. The health surrogate is a 3×3 table over patient regime (healthy/off-treatment, intermediate, severe) and action-intensity regime (none, one treatment, at least two treatments). Details are in Appendix A.2.

6.3 Experiment 3: PointMaze Medium

Experiment 3 extends the framework to a continuous-state and continuous-action control problem. We use Gymnasium-Robotics PointMaze Medium. The environment state is

$$s_t = (x_t, y_t, v_t^x, v_t^y) \in \mathbb{R}^4,$$

where (x_t, y_t) is the agent's continuous position and (v_t^x, v_t^y) is its velocity. The action is a two-dimensional force

$$a_t = (a_t^x, a_t^y) \in [-1, 1]^2.$$

Although the state space is continuous, trajectories are recorded at the simulator's primitive time steps. Thus a trajectory is stored as a finite sequence

$$\tau_i = \{(s_{i,t}, a_{i,t}, s_{i,t+1})\}_{t=0}^{L_i-1},$$

where $L_i \leq H$. In our experiments, the maximum horizon is $H = 300$ environment steps. We fix the start cell to $(1,1)$ and the goal cell to $(6,6)$ in the official medium maze.

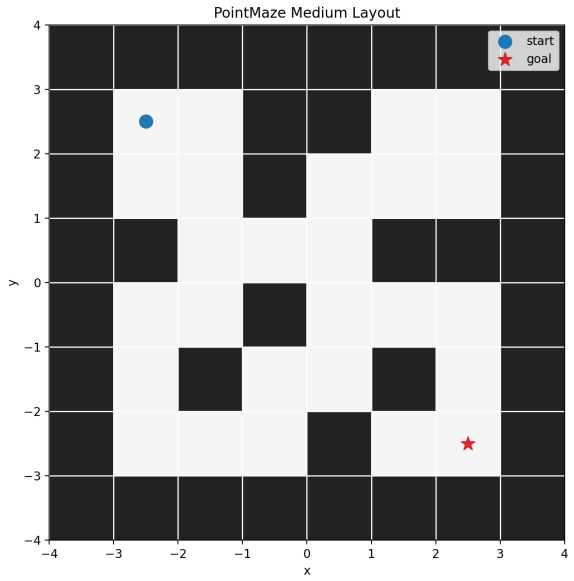
The observed label is a sparse binary trajectory-level outcome:

$$Y_i = \mathbf{1}\{\text{trajectory } i \text{ reaches the endpoint within } H \text{ steps}\}. \tag{27}$$

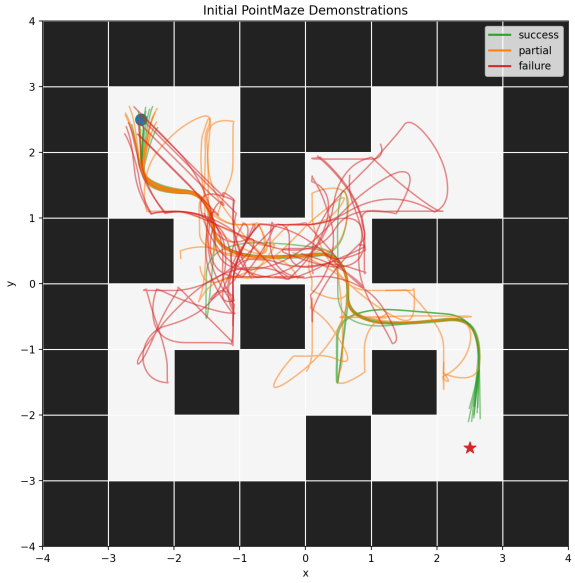
Therefore each trajectory provides only one aggregate label. As in the tabular experiments, we train primitive pseudo-reward models by requiring their predictions to add up to the observed trajectory label:

$$\hat{Y}_i = \sum_{t=0}^{L_i-1} \hat{r}(s_{i,t}, a_{i,t}) \approx Y_i. \tag{28}$$

This setting tests whether the proposed reward-recovery framework can turn sparse trajectory-level success labels into useful primitive pseudo-rewards for continuous-control policy learning.



(a) PointMaze layout with fixed start and goal.



(b) Initial dataset: successful, partial, and failed trajectories.

Figure 1: PointMaze setup. The initial dataset contains 30 trajectories: 10 successful trajectories, 10 partial demonstrations, and 10 exploratory failures.

For reward recovery, we use one flexible reference model and two structured surrogate models. The reference model is a DNN reward estimator with input

$$(x, y, v^x, v^y, a^x, a^y).$$

It plays the same role as the ridge reference estimator in the tabular experiments: it is flexible and intended to reduce asymptotic bias, but it may have high variance when the number of labeled trajectories is small.

The first surrogate is a box-region reward model. It ignores velocity and action and assigns one scalar reward parameter to each free 1×1 maze cell. If $c(s)$ denotes the maze cell containing the continuous position (x, y), then

$$\hat{r}_{\text{box}}(s, a) = \theta_{c(s)}.$$

This surrogate is intentionally coarse: it can capture which regions of the maze are associated with successful trajectories, but it cannot represent fine continuous dynamics or action-dependent effects.

The second surrogate is a heuristic-feature neural model. It first maps each state-action pair to three hand-designed geometric features:

$$\phi_{\text{heur}}(s, a) = \big(d_{\text{goal}}(s), n_{\text{wall}}(s, g), d_{\text{next wall}}(s, a)\big).$$

Here $d_{\text{goal}}(s)$ is the Euclidean distance from the current position to the goal, $n_{\text{wall}}(s, g)$ is the number of wall cells intersected by the straight line from the current position to the goal, and $d_{\text{next wall}}(s, a)$ is the ray-cast distance to the next wall if the current action direction is maintained. A small neural network maps these three features to a scalar pseudo-reward. This surrogate encodes simple geometric knowledge about maze navigation while remaining much lower-capacity than the DNN reference model.

The four methods are Delayed SAC, DNN-reference SAC, Adaptive MSE-SAC, and Disagreement-SAC. Delayed SAC is the direct model-free baseline: it uses reward zero on all non-final transitions and places the trajectory label on the final transition. DNN-reference SAC trains only the flexible neural reference reward model and relabels replay-buffer transitions with its predicted pseudo-reward. Adaptive MSE-SAC and Disagreement-SAC use the DNN reference, the box-region surrogate, and the heuristic-feature surrogate. Since tabular count-based UCB is not well-defined in continuous state-action space, SAC's entropy-regularized objective provides generic exploration, while MSE and disagreement terms provide reward-space optimism. Detailed baseline definitions are provided in Appendix A.3.

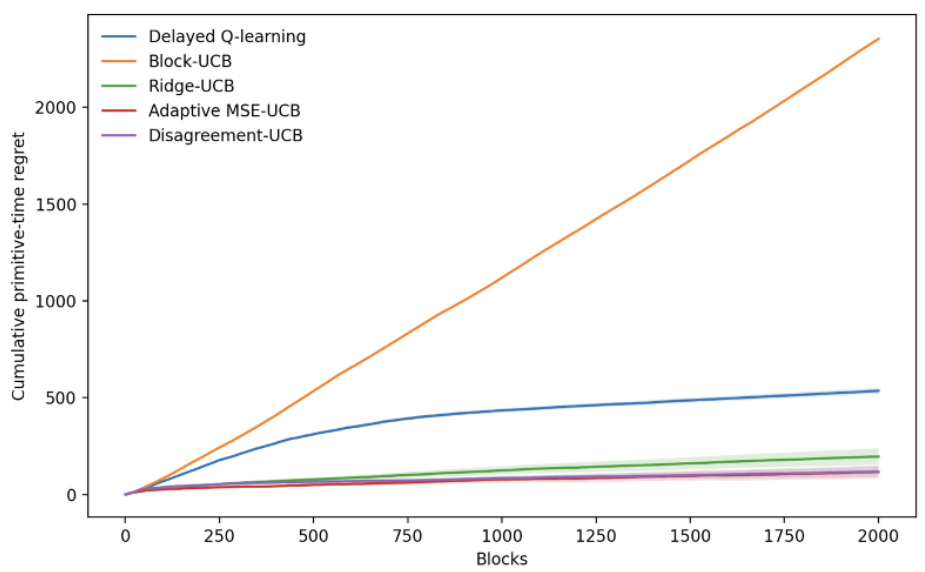


Figure 2: Experiment 1: cumulative primitive-time regret in the Toy Advertiser MDP. Reward-recovery methods substantially outperform direct delayed-feedback and block-level baselines. Adaptive MSE-UCB and Disagreement-UCB are the strongest methods, with Disagreement-UCB competitive in regret and best in policy/reward recovery diagnostics.

7 Results

7.1 Quantitative Evaluation: Performance and Sample Efficiency

The quantitative analysis focuses on policy performance: cumulative regret in the tabular and Sepsis experiments, and evaluation success rate in PointMaze. The main hypothesis is not that disagreement must dominate every metric in every environment, but that explicit reward or pseudo-reward recovery improves delayed-feedback learning, and that surrogate-assisted reward recovery gives additional gains beyond a flexible reference model alone.

Experiment 1: reward recovery under true delayed aggregation. The Toy Advertiser MDP provides the cleanest test of the delayed aggregated reward setting because the observed block label is exactly a noisy sum of primitive rewards. Figure 2 shows that methods which explicitly recover primitive rewards substantially outperform methods that do not. Block-UCB performs worst because it treats each length-T action sequence as a macro-action, causing the effective action space to grow exponentially with the delay horizon. Delayed Q-learning improves over Block-UCB but remains much worse than the reward-recovery methods, indicating that directly propagating the entire block reward to the final primitive step is not sufficient for efficient credit assignment.

Among reward-recovery methods, Ridge-UCB already achieves a large improvement over both delayed model-free learning and block-level planning, confirming that estimating a primitive pseudo-reward table from aggregate labels is useful. Adaptive MSE-UCB and Disagreement-UCB further improve performance by combining the ridge reference estimator with structured surrogate models. Table 1 reports the final summary. Adaptive MSE-UCB has the lowest final and area-under-curve regret, while Disagreement-UCB has comparable final regret and achieves the lowest policy error and mixed reward MSE. This supports the core hypothesis that surrogate-assisted reward recovery improves learning; it also suggests that disagreement is especially useful for reward-structure recovery and policy identification, even when its regret advantage over MSE-based optimism is modest in this simple tabular setting.

Experiment 2: pseudo-reward learning under structured delayed feedback. The Sepsis experiment tests whether the same reward-recovery idea remains useful in a larger structured MDP. We evaluate two feedback modes. In the additive mode, the observed label is the cumulative clinical utility over the treatment window, matching the delayed aggregated reward assumption. In the terminal mode, the observed label is only the final health outcome. The latter setting is formally misspecified for additive reward recovery, so the learned primitive reward should be interpreted as a pseudo-reward for credit assignment rather than as the true clinical reward.

Figure 3 reports the current Sepsis regret curves for the terminal mode. The important conceptual point is that reward-recovery methods improve over random exploration and delayed model-free learning, indicating that explicit pseudo-reward

Table 1: Experiment 1 final quantitative summary over 20 seeds. Lower is better for regret, policy error, and reward MSE. Block-UCB and delayed Q-learning do not learn a meaningful mixed reward estimate, so their reward-MSE entries are not interpreted as estimator diagnostics.

Method	Final regret	AUC regret	Policy error	Mixed reward MSE
Block-UCB	2353.91 ± 7.11	$2.27 \times 10^6 \pm 6.01 \times 10^3$	0.965 ± 0.011	–
Delayed Q-learning	535.71 ± 15.33	$7.62 \times 10^5 \pm 1.75 \times 10^4$	0.455 ± 0.033	–
Ridge-UCB	195.42 ± 42.07	$2.37 \times 10^5 \pm 4.14 \times 10^4$	0.320 ± 0.049	0.00212 ± 0.00032
Adaptive MSE-UCB	116.04 ± 33.24	$1.44 \times 10^5 \pm 3.26 \times 10^4$	0.320 ± 0.050	0.00277 ± 0.00035
Disagreement-UCB	118.11 ± 25.89	$1.62 \times 10^5 \pm 2.39 \times 10^4$	0.280 ± 0.052	0.00149 ± 0.00014

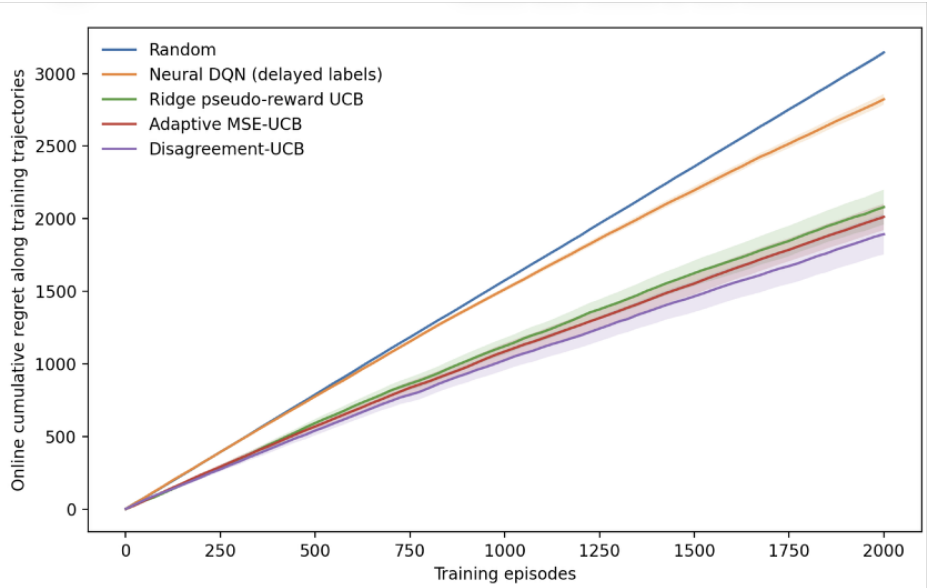


Figure 3: Experiment 2: online cumulative regret in the Sepsis simulator. The experiment evaluates both true additive feedback and terminal-outcome feedback. The key use of this environment is to test whether pseudo-reward recovery remains useful in a structured MDP even when the observed label is not a true primitive reward aggregate.

recovery can provide useful intermediate learning signals in a structured environment. We use this experiment to separate two contributions. First, when the delayed label is truly additive, reward recovery targets the true primitive reward. Second, when the delayed label is a terminal outcome, the same machinery can still learn a useful pseudo-reward for planning.

Experiment 3: continuous control with trajectory-level sparse feedback. The PointMaze experiment extends the idea to continuous state and action spaces. Because tabular value iteration and state-action counts no longer apply, we use SAC as the policy optimizer. Delayed SAC receives the trajectory label only at the final transition, while the reward-recovery methods train SAC on relabeled primitive transitions.

Figure 4 reports evaluation success rate as a function of the number of online trajectories. The curves are noisier than in the tabular experiments because SAC training, continuous dynamics, and sparse endpoint labels introduce additional variance. Nevertheless, the reward-recovery methods begin to improve once enough online data are collected, while Delayed SAC learns more slowly from the same sparse endpoint labels. The DNN-reference baseline tests whether a flexible neural reward model alone is sufficient. Adaptive MSE-SAC and Disagreement-SAC test whether adding structured surrogate models and local uncertainty/disagreement signals improves the learned reward used by SAC. The observed improvement of the adaptive and disagreement-based methods over the pure delayed-reward baseline supports the hypothesis that even in continuous control, trajectory-level sparse labels can be converted into useful primitive pseudo-rewards for policy optimization.

Across all three experiments, the most robust quantitative pattern is that direct delayed-reward baselines are less sample-efficient than methods that recover primitive pseudo-rewards. The strongest evidence comes from the Toy Advertiser MDP, where the aggregated-reward assumption is exactly correct, and from PointMaze, where the observed reward is sparse and trajectory-level rather than dense. These results support the central claim that reward recovery is useful both for true delayed

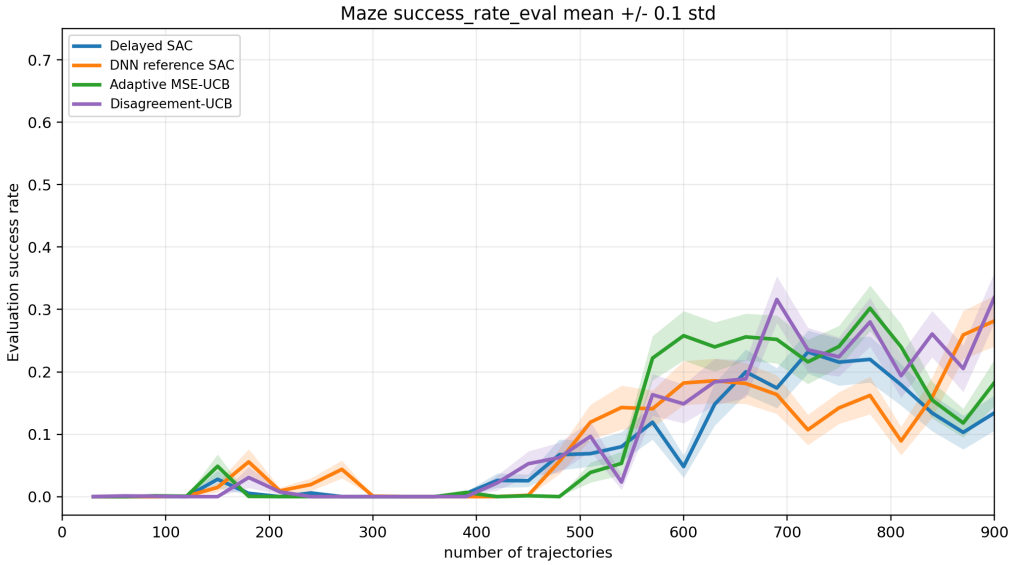


Figure 4: Experiment 3: PointMaze evaluation success rate. The continuous setting is noisier, but reward-recovery and reward-space optimism improve learning relative to direct delayed SAC in several parts of training.

aggregates and for pseudo-reward learning from sparse outcome labels.

7.2 Qualitative Analysis: Reward Structure, Model Trust, and Exploration Mechanism

The qualitative analysis focuses on mechanism rather than aggregate performance. While the quantitative results answer whether a method improves regret or success rate, the qualitative results explain why the improvement occurs and when it may fail.

Reward structure recovery in the Toy Advertiser MDP. Figure 5 compares the true reward table with the ridge estimate, the linear surrogate, the threshold surrogate, the mixed reward, and the learned ridge weight. The true reward has a nontrivial state-action structure: aggressive advertising is valuable in some intermediate states but costly or inefficient elsewhere. The ridge estimator is flexible and can represent the true reward table, but it has higher variance early in training because each state-action pair must be learned separately from aggregate labels. The linear and threshold surrogates impose lower-dimensional structure and therefore provide smoother low-variance estimates, but they are misspecified.

The mixed reward combines these complementary signals. The heatmaps show that the mixed estimate preserves the broad structure of the true reward while avoiding some high-variance artifacts of the ridge estimate. The model-weight heatmap provides a local interpretation of the adaptive mixture: the algorithm does not choose a single global reward model, but instead assigns different reliability weights across the state-action space. This supports the main design principle of the method: surrogate models should not replace the reference estimator; they should be used where they are locally credible.

Disagreement as a model-discrimination signal. Disagreement is not merely another uncertainty bonus. It is intended to identify regions where the flexible reference estimator and a structured surrogate imply different primitive reward assignments. In such regions, additional samples are useful because they help the learner decide whether the surrogate's inductive bias is locally reliable or misspecified. This is qualitatively different from Adaptive MSE-UCB, which explores where the currently best model has high estimated MSE. The Toy Advertiser diagnostics visualize this mechanism: large disagreement appears in state-action regions where the low-dimensional surrogate and ridge estimator encode different reward extrapolations. Exploring such regions helps explain why Disagreement-UCB can obtain lower reward MSE and policy error even when its cumulative regret is close to Adaptive MSE-UCB.

Pseudo-reward interpretation in Sepsis. The Sepsis experiment highlights an important distinction between true reward recovery and pseudo-reward recovery. In additive mode, the learned primitive reward can be interpreted as an estimator of expected clinical utility. In terminal mode, however, the observed label is only the final health outcome. The learned primitive reward is therefore not a literal clinical reward at each step; it is a pseudo-reward that assigns credit to intermediate state-action

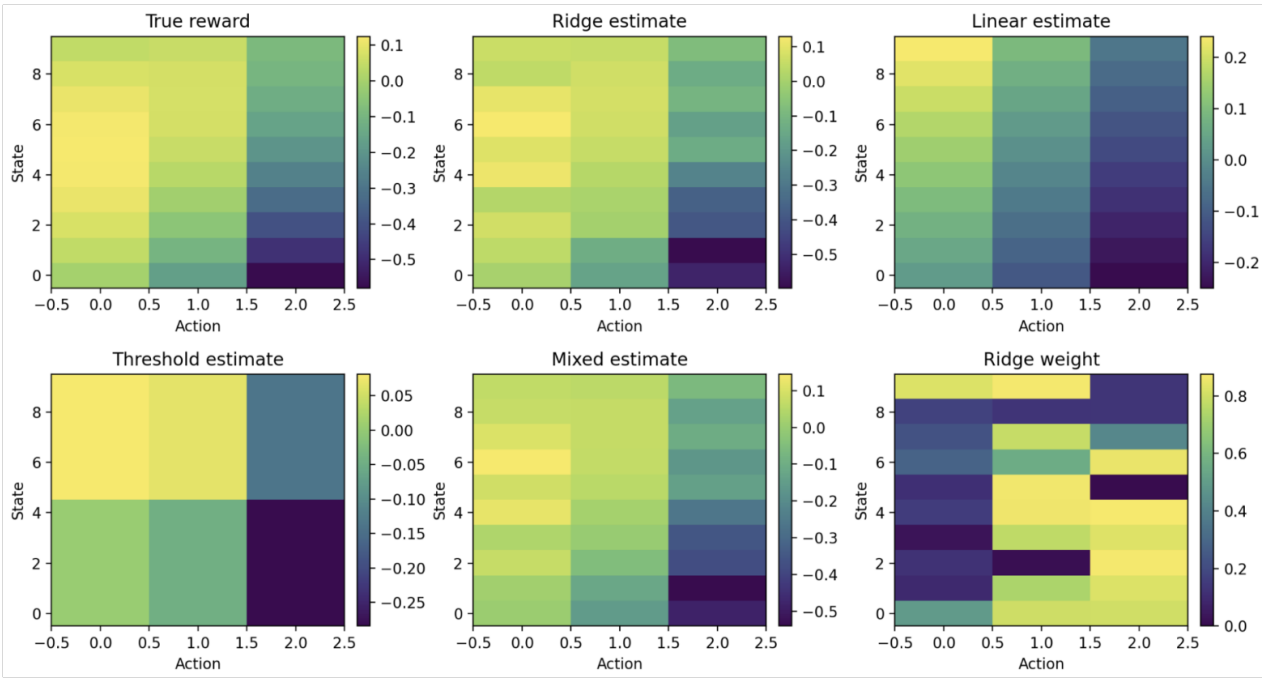


Figure 5: Experiment 1 qualitative diagnostics: true reward, recovered reward estimates, mixed reward, and ridge weight. The mixed estimate preserves the qualitative structure of the true reward while the weights vary across state-action regions.

pairs that are predictive of the final outcome. This distinction is important for the paper's contribution. The method is not limited to settings where the observed signal is exactly a sum of primitive rewards. It can also be used as a credit-assignment heuristic when the observed reward is sparse, delayed, or terminal.

Initial demonstrations and learned behavior in PointMaze. Figure 1 provides the qualitative diagnostic for PointMaze. The maze layout fixes the start in the upper-left region and the goal in the lower-right region. The initial dataset contains three types of trajectories: successful demonstrations, partial demonstrations, and failed exploratory trajectories. Successful trajectories ensure that the dataset contains positive endpoint labels, while partial and failed trajectories provide coverage of useful corridors and off-path regions. This design prevents the reward model from seeing only all-zero labels while avoiding a pure imitation-learning setup.

The qualitative trajectory overlays show how methods differ in the behavior they induce. Delayed SAC receives almost no informative primitive feedback and therefore relies heavily on sparse terminal learning. In contrast, the reward-recovery methods relabel primitive transitions using learned pseudo-rewards, giving SAC denser learning signals. The learned reward and model-weight heatmaps can be used to diagnose whether the algorithm places reward mass near useful maze regions, whether the box surrogate captures coarse spatial structure, and whether the heuristic surrogate contributes in regions where geometric features are informative. These diagnostics are essential because in continuous control, final success rate alone does not reveal whether the reward model has learned a meaningful credit-assignment structure or merely overfit to a small set of successful trajectories.

Overall mechanism. Taken together, the qualitative results support the same mechanism across environments. The reference estimator provides flexibility and guards against surrogate misspecification. The surrogate models provide low-variance structural guidance when data are limited. Adaptive weighting determines which estimator is locally reliable. Disagreement identifies regions where model assumptions conflict and where additional exploration is informative. This mechanism explains why the approach improves over direct delayed-reward baselines and why its benefit is not restricted to exactly additive reward observations.

8 Discussion

The experiments suggest that the method is clearest when the delayed label is truly additive, as in the Toy Advertiser MDP. In this case, ridge reward recovery is statistically well-aligned with the data-generating process, and surrogates provide useful

low-variance structure. The Sepsis terminal-feedback and PointMaze experiments are more subtle. Their labels are not literally sums of primitive rewards, yet fitting pseudo-additive rewards can still help policy learning by creating a dense credit-assignment signal. This is important because many real applications provide endpoint outcomes rather than decomposable rewards.

The main limitation is uncertainty estimation. In tabular experiments, bootstrap and count bonuses provide simple MSE and exploration proxies. In continuous PointMaze, full bootstrap is computationally expensive, so the implementation uses analytic variance for the box surrogate and MC-dropout for neural reward models. These are practical proxies, not calibrated confidence intervals. Another limitation is that the continuous experiment uses SAC as a black-box policy optimizer; reward-space optimism may interact with SAC entropy and critic extrapolation in ways not captured by the tabular theory.

9 Conclusion

This project studies online credit assignment under delayed aggregate and pseudo-aggregate feedback. The central message is that learning a primitive pseudo-reward can substantially improve policy learning when direct delayed-feedback RL is inefficient. Structured surrogate models provide useful low-variance inductive bias, while a flexible reference estimator protects against misspecification. Disagreement between these models is both interpretable and useful for exploration. The method performs well in a tabular advertiser MDP, improves learning in a structured Sepsis simulator, and extends to continuous PointMaze through SAC-based reward relabeling. Future work should develop better uncertainty estimation for neural reward recovery and test the approach in higher-dimensional continuous-control environments.

10 Team Contributions

Haozhan Gao developed the research idea, implemented the experiments, ran the analyses, prepared the poster, and wrote the final report. Jason Meng is responsible for the theoretical analysis part. Jose Blanchet provided advising and feedback on the research direction and presentation.

References

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of machine learning research*, 3(Nov): 397–422, 2002.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International conference on machine learning*, pages 263–272. PMLR, 2017.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Delay and cooperation in nonstochastic bandits. *Journal of Machine Learning Research*, 20(17):1–38, 2019.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Sam Michael Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 433–440. IFAAMAS, 2012.

Marek Grzes and Daniel Kudenko. Learning shaping rewards in model-based reinforcement learning. In *Proc. AAMAS 2009 Workshop on Adaptive Learning Agents*, volume 115, page 30, 2009.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International conference on machine learning*, pages 1453–1461. PMLR, 2013.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Maja J Mataric. Reward functions for accelerated learning. In *Machine learning proceedings 1994*, pages 181–189. Elsevier, 1994.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287. Citeseer, 1999.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models, 2019. URL https://arxiv.org/abs/1905.05824.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*, pages 4105–4113. PMLR, 2018.

Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *ICML*, volume 98, pages 463–471, 1998.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020.

Satinder P Singh and Richard S Sutton. Reinforcement learning with replacing eligibility traces. *Machine learning*, 22(1): 123–158, 1996.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. *arXiv preprint arXiv:1706.09186*, 2017.

Eric Wiewiora. Potential-based shaping and q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19:205–208, 2003.

A Detailed Baseline Algorithms

This appendix gives the precise baseline and ablation definitions used in the experiments. The main paper keeps the experimental setup concise; the details here clarify exactly which component each baseline removes.

A.1 Experiment 1 baselines

All Experiment 1 methods observe the same stream of primitive states and actions but only receive a noisy block return after every $T = 7$ primitive steps.

Delayed Q-learning. This is the direct model-free delayed-feedback baseline. It does not recover primitive rewards. For a block with observed label Y_i, it defines artificial rewards

$$r_t^{\text{obs}} = 0 \quad (t < T-1), \qquad r_{T-1}^{\text{obs}} = Y_i. \tag{29}$$

It then performs tabular Q-learning with ϵ-greedy exploration. This baseline tests whether direct temporal-difference learning can solve the credit-assignment problem without reward recovery.

Block-UCB. This is the direct block-level planning baseline. It treats a length-T primitive action sequence $u = (a_0, \dots, a_{T-1}) \in \mathcal{A}^T$ as a macro-action. It estimates a block-level reward $\hat{R}(s, u)$ and block transition $\hat{P}(s' \mid s, u)$ from observed blocks and plans with a block-level UCB bonus. This baseline avoids primitive reward recovery, but its action space grows as $|\mathcal{A}|^T$. It isolates the cost of solving the delayed problem at the block level.

Ridge-UCB. This is the flexible reference-only reward recovery baseline. It fits the tabular ridge estimator in (4), estimates transition probabilities from primitive transition counts, and plans in the original MDP using

$$R_k^{\text{ridge}}(s, a) = \tilde{r}_k(s, a) + b_{\text{count},k}(s, a) + b_{\text{ridgeMSE},k}(s, a). \tag{30}$$

This baseline tests whether model-agnostic reward recovery is sufficient without surrogate structure.

Adaptive MSE-UCB. This method uses the same candidate reward models as the full algorithm: ridge, linear surrogate, and threshold surrogate. It computes local MSE proxies, forms the relative-MSE mixed reward $\bar{r}_k$, and adds an optimism bonus based on the MSE of the locally best model:

$$R_k^{\text{MSE}}(s, a) = \bar{r}_k(s, a) + b_{\text{count},k}(s, a) + c_{\text{mse}} \frac{\sqrt{\widehat{\text{MSE}}_{j^*}(s, a)}}{\sqrt{\lambda + N_k(s, a)}}. \tag{31}$$

This is the key ablation for the full algorithm because it includes surrogate modeling and adaptive mixing but removes disagreement-based exploration.

Disagreement-UCB. This is the full method. It uses the same model family and the same mixed reward as Adaptive MSE-UCB. If the reference model is locally best, it explores according to reference uncertainty. If a surrogate is locally best, it explores according to reference-surrogate disagreement:

$$e_k(s, a) = \begin{cases} \sqrt{\widehat{\text{MSE}}_{\text{ridge}}(s, a)}, & j^*(s, a) = \text{ridge}, \\ \sqrt{\widehat{\Delta}^2_{j^*}(s, a)}, & j^*(s, a) \in \{\text{linear}, \text{threshold}\}. \end{cases} \tag{32}$$

It plans with $\bar{r}_k + b_{\text{count}} + c_{\text{dis}} e_k / \sqrt{\lambda + N_k}$.

A.2 Experiment 2 baselines

Experiment 2 uses the same comparison logic, but in an episodic Sepsis simulator with horizon $T = 5$ and either additive or terminal feedback.

Random policy. This lower-bound baseline chooses uniformly among the eight treatment combinations. It checks that the environment is nontrivial and gives a reference level for unstructured exploration.

Delayed DQN. This is the model-free delayed-label baseline. It receives artificial reward zero at nonfinal steps and the observed window label at the final step. Its input is the state representation plus a time index, and its output is a Q-value for each treatment action. It tests whether a neural value-based method can solve the delayed credit-assignment problem without explicit pseudo-reward recovery.

Ridge pseudo-reward UCB. This baseline fits a tabular pseudo-reward over Sepsis state-action pairs using the same aggregate or terminal labels. In additive mode, this is aligned with the data-generating structure. In terminal mode, it is intentionally a pseudo-reward recovery method, not a true primitive reward estimator. Planning uses finite-horizon value iteration with the recovered reward and empirical transition model.

Adaptive MSE-UCB. This method fits ridge, linear clinical surrogate, and health-regime surrogate models. It estimates local MSE proxies, forms an adaptive mixed reward, and adds a best-MSE exploration bonus. It controls for the benefit of adaptive reward mixing without disagreement-based exploration.

Disagreement-UCB. This is the full Sepsis method. It uses the same reward model family and mixed reward as Adaptive MSE-UCB, but when a surrogate is locally best it uses disagreement with ridge as the exploration signal. This tests whether the additional exploration value comes from disagreement rather than from the mere presence of clinical surrogate models.

A.3 Experiment 3 baselines

Experiment 3 is continuous, so SAC replaces value iteration. All methods start from the same 30-trajectory initialization and then collect new trajectories online.

Delayed SAC. This is the direct continuous-control baseline. It assigns reward zero to all nonfinal transitions and the trajectory label $Y_i \in \{0, 1\}$ to the final transition. SAC is trained on these sparse delayed rewards. This baseline tests whether standard off-policy continuous-control learning can solve the task without reward recovery.

DNN-reference SAC. This is the continuous analog of Ridge-UCB without UCB. It trains only the flexible DNN reference reward model from trajectory-level labels and relabels replay-buffer transitions with $\tilde{r}_\psi(s, a)$. It tests whether a flexible learned pseudo-reward alone is enough, without structured surrogates or disagreement.

Adaptive MSE-SAC. This method fits the DNN reference model, the box-region surrogate, and the heuristic-feature NN surrogate. It computes local uncertainty/MSE proxies, forms the relative-MSE mixed reward, and adds a reward-space bonus based on the best local MSE proxy:

$$R_k^{\mathrm{MSE}}(s, a) = \bar{r}_k(s, a) + \beta_{\mathrm{mse}} \sqrt{\min_j \widehat{\mathrm{MSE}}_{j,k}(s, a)}. \tag{33}$$

There is no tabular count bonus because $N(s, a)$ is not defined in continuous state-action space.

Disagreement-SAC. This is the full continuous method. It uses the same DNN reference, box surrogate, heuristic surrogate, and adaptive mixed reward as Adaptive MSE-SAC. If the reference model is locally best, it uses reference uncertainty. If a surrogate is locally best, it uses reference-surrogate disagreement:

$$R_k^{\mathrm{Dis}}(s, a) = \bar{r}_k(s, a) + \beta_{\mathrm{dis}} e_k(s, a). \tag{34}$$

SAC is then trained on replay-buffer transitions relabeled with R_k^{Dis}. Online data collection uses the stochastic SAC actor, while evaluation uses the deterministic mean action and is not added to the replay buffer.

B Disagreement-Seeking Algorithms

Algorithm 1 Tabular Disagreement-UCB

Initialize dataset $\mathcal{D}_0$, transition counts $N_0(s, a, s')$, and visit counts $N_0(s, a)$.
for $k = 0, 1, 2, \ldots$ **do**
  Fit the reference reward model $\hat{r}_{\mathrm{ref},k}$ on $\mathcal{D}_k$.
  Fit each surrogate reward model $\hat{r}_{m,k}$, $m \in \mathcal{M}$, on $\mathcal{D}_k$.
  for $b = 1$ to B **do**
    Resample trajectories/windows from $\mathcal{D}_k$ with replacement to form $\mathcal{D}_k^{(b)}$.
    Refit the reference model $\hat{r}_{\mathrm{ref},k}^{(b)}$ on $\mathcal{D}_k^{(b)}$.
    Refit each surrogate model $\hat{r}_{m,k}^{(b)}$ on $\mathcal{D}_k^{(b)}$.
  end for
  Estimate local MSE proxies $\widehat{\mathrm{MSE}}_{j,k}(s, a)$ for all $j \in \mathcal{M}^+$.
  Compute adaptive weights $w_{j,k}(s, a)$ using the relative-MSE softmax in (14).
  Form the mixed reward

$$\bar{r}_k(s, a) = \sum_{j \in \mathcal{M}^+} w_{j,k}(s, a) \hat{r}_{j,k}(s, a).$$

  Let

$$j_k^\star(s, a) \in \arg\min_{j \in \mathcal{M}^+} \widehat{\mathrm{MSE}}_{j,k}(s, a).$$

  Define the disagreement exploration signal

$$e_k(s, a) = \begin{cases} \sqrt{\widehat{\mathrm{MSE}}_{\mathrm{ref},k}(s, a)}, & j_k^\star(s, a) = \mathrm{ref}, \\ \sqrt{\widehat{\Delta}^2_{j_k^\star,k}(s, a)}, & j_k^\star(s, a) \in \mathcal{M}. \end{cases}$$

  Estimate the transition model $\hat{P}_k(s' \mid s, a)$ from transition counts.
  Define the optimistic planning reward

$$r_k^{\mathrm{plan}}(s, a) = \bar{r}_k(s, a) + \frac{c_{\mathrm{base}}}{\sqrt{\lambda + N_k(s, a)}} + \frac{c_{\mathrm{dis}} e_k(s, a)}{\sqrt{\lambda + N_k(s, a)}}.$$

  Compute policy π_k by value iteration using $(\hat{P}_k, r_k^{\mathrm{plan}})$.
  Execute π_k for one delayed-feedback window of length T.
  Observe the aggregate label Y_k and the state-action trajectory, but not primitive rewards.
  Append the new window to $\mathcal{D}_k$ and update $N_k(s, a, s')$ and $N_k(s, a)$.
end for

Algorithm 2 Continuous Disagreement-SAC

Initialize trajectory dataset $\mathcal{D}_0$ and replay buffer $\mathcal{B}_0$ from initial trajectories.
Initialize SAC actor and critic networks.
for $k = 0, 1, 2, \ldots$ **do**
  Fit the DNN reference reward model $\hat{r}_{\mathrm{ref},k}$ on trajectory-level labels in $\mathcal{D}_k$.
  Fit each continuous surrogate reward model $\hat{r}_{m,k}$, $m \in \mathcal{M}$, on $\mathcal{D}_k$.
  Estimate local uncertainty/MSE proxies $\widehat{\mathrm{MSE}}_{j,k}(s,a)$ for all $j \in \mathcal{M}^+$.
  Compute adaptive weights $w_{j,k}(s,a)$ using the relative-MSE softmax in (14).
  Form the mixed primitive pseudo-reward

$$\bar{r}_k(s,a) = \sum_{j \in \mathcal{M}^+} w_{j,k}(s,a)\hat{r}_{j,k}(s,a).$$

  Let

$$j_k^\star(s,a) \in \arg\min_{j \in \mathcal{M}^+} \widehat{\mathrm{MSE}}_{j,k}(s,a).$$

  Define the reward-space disagreement signal

$$e_k(s,a) = \begin{cases} \sqrt{\widehat{\mathrm{MSE}}_{\mathrm{ref},k}(s,a)}, & j_k^\star(s,a) = \mathrm{ref}, \\ \left|\hat{r}_{\mathrm{ref},k}(s,a) - \hat{r}_{j_k^\star,k}(s,a)\right|, & j_k^\star(s,a) \in \mathcal{M}. \end{cases}$$

  Define the primitive reward used by SAC

$$\hat{r}_k^{\mathrm{plan}}(s,a) = \bar{r}_k(s,a) + \beta_{\mathrm{dis}} e_k(s,a).$$

  Relabel each replay transition $(s_t, a_t, s_{t+1}) \in \mathcal{B}_k$ with

$$\tilde{r}_t^{(k)} = \hat{r}_k^{\mathrm{plan}}(s_t, a_t).$$

  Train SAC for a fixed number of gradient steps on relabeled transitions

$$(s_t, a_t, \tilde{r}_t^{(k)}, s_{t+1}).$$

  Evaluate the deterministic mean-action SAC policy on fresh rollouts for logging only.
  Collect new online trajectories using the stochastic SAC policy $a_t \sim \pi_k(\cdot \mid s_t)$.
  Observe one trajectory-level label Y_i for each new trajectory, but not primitive rewards.
  Append new trajectories to $\mathcal{D}_k$ and their transitions to $\mathcal{B}_k$.
end for

C Theoretical Appendix

C.1 Definitions

C.1.1 Diameter and span

Let D denote the diameter of the communicating MDP:

$$D := \max_{s,s' \in \mathcal{S}} \min_{\pi} \mathbb{E}_\pi[\tau(s' \mid s)],$$

where $\tau(s' \mid s)$ is the (random) hitting time of s' starting from s under policy π. For any function $h : \mathcal{S} \to \mathbb{R}$, define its span $span(h) := \max_s h(s) - \min_s h(s)$. A standard fact in communicating average-reward MDPs is that there exists an optimal bias function h^* with $span(h^*) \le D$ (see, e.g., standard average-reward DP theory Auer et al. (2008)).

C.2 Proof of Theorem 1

We prove the theorem by combining (i) concentration for transitions, (ii) concentration for ridge reward recovery with $\sqrt{T}$ dependence, (iii) optimism, and (iv) a standard UCRL2 regret decomposition plus a counting argument.

Step 1: Transition confidence sets. Let $N_e(s,a)$ be the number of primitive visits to (s,a) prior to epoch e, and let $N_e(s,a,s')$ be the corresponding transition count. Define the empirical kernel

$$\hat{P}_e(s' \mid s,a) := \frac{N_e(s,a,s')}{\max\{1, N_e(s,a)\}}.$$

Fix $\delta \in (0,1)$. For each (s,a) define

$$b_e^p(s,a) \;:=\; \sqrt{\frac{2\log\!\left(\frac{2SA\,N}{\delta}\right)}{\max\{1, N_e(s,a)\}}}. \tag{35}$$

By a standard multinomial deviation bound (e.g., Weissman et al. type inequality) and a union bound over (s,a) and all epochs up to time N, with probability at least $1-\delta/2$,

$$\forall e, \forall (s,a): \quad \|P(\cdot \mid s,a) - \hat{P}_e(\cdot \mid s,a)\|_1 \le b_e^p(s,a). \tag{36}$$

Denote this event by $\mathcal{E}_P$.

Step 2: Ridge reward recovery confidence with $\sqrt{T}$. Let M_e be the number of completed blocks before epoch e. Stack the block features into $X_e \in \mathbb{R}^{M_e \times d}$ with rows $x_0^\top, \dots, x_{M_e-1}^\top$, and responses into $y_e \in \mathbb{R}^{M_e}$. Then the ridge estimator (4) satisfies

$$\tilde{\theta}_e \;=\; (X_e^\top X_e + \lambda I)^{-1} X_e^\top y_e.$$

Define $V_e := X_e^\top X_e + \lambda I$. Let $\epsilon = (\epsilon_0, \dots, \epsilon_{M_e-1})^\top$ and note $y_e = X_e\theta^* + \epsilon$. Then

$$\tilde{\theta}_e - \theta^* \;=\; V_e^{-1} X_e^\top \epsilon.$$

We will use a standard self-normalized concentration inequality for linear regression with sub-Gaussian noise: with probability at least $1-\delta/2$, for all epochs e,

$$\|X_e^\top \epsilon\|_{V_e^{-1}} \;\le\; \sigma\sqrt{2\log\!\Big(\frac{\det(V_e)^{1/2}}{\det(\lambda I)^{1/2}} \cdot \frac{2}{\delta}\Big)}. \tag{37}$$

Denote this event by $\mathcal{E}_R$. Assumption 1 yields a bound on $\det(V_e)$ as follows. Since $\mathrm{tr}(X_e^\top X_e) = \sum_{m=0}^{M_e-1} \|x_m\|_2^2 \le M_e \kappa T$, we have

$$\det(V_e) = \det(\lambda I + X_e^\top X_e) \;\le\; \Big(\lambda + \frac{\mathrm{tr}(X_e^\top X_e)}{d}\Big)^d \;\le\; \Big(\lambda + \frac{M_e \kappa T}{d}\Big)^d.$$

Plugging into (37), there exists a logarithmic factor

$$\mathcal{L}_e := \log\!\Big(\Big(1 + \frac{M_e\kappa T}{\lambda d}\Big)^{d/2} \cdot \frac{2}{\delta}\Big)$$

such that on $\mathcal{E}_R$,

$$\|\tilde{\theta}_e - \theta^*\|_{V_e} \;\le\; \sigma\sqrt{2\mathcal{L}_e}. \tag{38}$$

Now fix a coordinate j corresponding to some (s,a). Let $e_j \in \mathbb{R}^d$ denote the j-th basis vector. Then

$$|\tilde{\theta}_e(j) - \theta^*(j)| \;=\; |e_j^\top(\tilde{\theta}_e - \theta^*)| \;\le\; \|\tilde{\theta}_e - \theta^*\|_{V_e} \cdot \|e_j\|_{V_e^{-1}} \;\le\; \sigma\sqrt{2\mathcal{L}_e} \cdot \sqrt{e_j^\top V_e^{-1} e_j}.$$

It remains to upper bound $e_j^\top V_e^{-1} e_j$. Note that $(V_e)_{jj} \;=\; \lambda + \sum_{m=0}^{M_e-1} x_m(j)^2 \;\ge\; \lambda + \sum_{m=0}^{M_e-1} x_m(j)$ since $x_m(j)$ are nonnegative integers and $x^2 \ge x$ for integers $x \ge 0$. But $\sum_m x_m(j) =: N_e(s,a)$ is exactly the number of primitive visits to (s,a) prior to epoch e. By Assumption 3, we have $V_e \succeq \eta\,\mathrm{diag}(V_e)$, hence $V_e^{-1} \preceq \eta^{-1}\mathrm{diag}(V_e)^{-1}$ and therefore

$$e_j^\top V_e^{-1} e_j \;\le\; \eta^{-1} e_j^\top \mathrm{diag}(V_e)^{-1} e_j \;=\; \frac{1}{\eta (V_e)_{jj}} \;\le\; \frac{1}{\eta(\lambda + N_e(s,a))}.$$

Therefore, on $\mathcal{E}_R$,

$$|\tilde{r}_e(s,a) - r(s,a)| \;=\; |\tilde{\theta}_e(j) - \theta^*(j)| \;\le\; b_e^{rec}(s,a) := \sigma\sqrt{\frac{2\mathcal{L}_e}{\eta(\lambda + N_e(s,a))}}. \tag{39}$$

Using the expression for $\mathcal{L}_e$ and the fact $M_e \le N/T$, we can simplify (absorbing logs into $\tilde{O}$):

$$b_e^{rec}(s,a) \;=\; \tilde{O}\left(\sigma\sqrt{\frac{\kappa T}{\eta(\lambda + N_e(s,a))}}\right). \tag{40}$$

Step 3: Reward UCB via recovery + disagreement. By triangle inequality,

$$|r(s,a) - \hat{r}_{\theta_e}(s,a)| \;\le\; |r(s,a) - \tilde{r}_e(s,a)| + |\tilde{r}_e(s,a) - \hat{r}_{\theta_e}(s,a)| \;\le\; b_e^{rec}(s,a) + \Delta_e(s,a).$$

By Assumption 2, $\Delta_e(s,a) \le \Delta_{\max}$. Define the reward bonus

$$b_e^r(s,a) \;:=\; b_e^{rec}(s,a) + \Delta_{\max}\sqrt{\frac{1}{\lambda + N_e(s,a)}}. \tag{41}$$

(The $\sqrt{1/(\lambda + N_e)}$ scaling for the disagreement term matches the usual UCB decay; since we assume only a crude uniform bound on Δ, this is conservative but convenient.) Then on $\mathcal{E}_R$, for all (s,a),

$$r(s,a) \;\le\; \hat{r}_{\theta_e}(s,a) + b_e^{rec}(s,a) + \Delta_{\max} \;\le\; \hat{r}_{\theta_e}(s,a) + b_e^r(s,a) + \Delta_{\max}\left(1 - \sqrt{\tfrac{1}{\lambda + N_e(s,a)}}\right).$$

For simplicity, we proceed with (41) and absorb constant slack into $\tilde{O}$.

Step 4: Optimism and epoch-wise regret decomposition. Let $\mathcal{E} := \mathcal{E}_P \cap \mathcal{E}_R$, which holds with probability at least $1 - \delta$ by union bound. Fix an epoch e and let t_e be its start time and t_{e+1} its end time (exclusive). Let $v_e(s,a)$ be the number of visits to (s,a) during epoch e. Let M_e^+ be an optimistic MDP chosen within the confidence sets defined by (35)–(41), and let π_e be an optimal stationary policy for M_e^+ with gain g_e^+ and bias function h_e^+ satisfying the average-reward optimality equations

$$g_e^+ + h_e^+(s) \;=\; \max_{a \in \mathcal{A}} \Big\{ r_e^+(s,a) + \sum_{s'} P_e^+(s' \mid s,a)\, h_e^+(s') \Big\}, \tag{42}$$

where $r_e^+(s,a)$ and $P_e^+(\cdot \mid s,a)$ are the optimistic reward/kernel used in M_e^+. By construction on $\mathcal{E}$, the true MDP (P,r) lies in the confidence set, hence the optimistic gain upper-bounds the optimal gain: $g_e^+ \ge g^*$ (standard UCRL optimism argument).

Now consider the per-step regret during epoch e:

$$\sum_{k=t_e}^{t_{e+1}-1} (g^* - r(s_k,a_k)) \;\le\; \sum_{k=t_e}^{t_{e+1}-1} (g_e^+ - r(s_k,a_k)).$$

We bound the RHS using the bias function h_e^+. From (42), for the action $a_k = \pi_e(s_k)$ chosen by the epoch policy,

$$g_e^+ + h_e^+(s_k) \;\ge\; r_e^+(s_k,a_k) + \sum_{s'} P_e^+(s' \mid s_k,a_k)\, h_e^+(s').$$

Rearrange and add/subtract true terms:

$$\begin{aligned} g_e^+ - r(s_k,a_k) \le \big(r_e^+(s_k,a_k) - r(s_k,a_k)\big) + \sum_{s'} \big(P_e^+(s' \mid s_k,a_k) - P(s' \mid s_k,a_k)\big) h_e^+(s') \\ + \Big(\sum_{s'} P(s' \mid s_k,a_k) h_e^+(s') - h_e^+(s_k)\Big). \end{aligned}$$

Summing over $k = t_e, \ldots, t_{e+1} - 1$, the last term telescopes into a martingale difference plus a bounded boundary term. Standard UCRL2 analysis yields

$$\sum_{k=t_e}^{t_{e+1}-1} (g_e^+ - r(s_k,a_k)) \;\le\; \sum_{s,a} v_e(s,a)\Big(r_e^+(s,a) - r(s,a)\Big) + (h_e^+) \sum_{s,a} v_e(s,a)\, \|P_e^+(\cdot \mid s,a) - P(\cdot \mid s,a)\|_1 + O(D), \tag{43}$$

where the $O(D)$ boundary term comes from the span of h_e^+ (details below) and can be absorbed. We now upper bound the two main sums using the confidence radii. On $\mathcal{E}$, by definition of the optimistic model inside the confidence sets,

$$r_e^+(s,a) - r(s,a) \;\le\; b_e^r(s,a), \qquad \|P_e^+(\cdot \mid s,a) - P(\cdot \mid s,a)\|_1 \;\le\; b_e^p(s,a).$$

Moreover, in communicating MDPs we can take ${}^(h_e^+) \le D$ for an optimal bias function of M_e^+ (and in any case, standard UCRL2 bounds ${}^(h_e^+)$ by a diameter-type constant). Thus (43) implies

$$\sum_{k=t_e}^{t_{e+1}-1} (g^* - r(s_k,a_k)) \;\le\; \sum_{s,a} v_e(s,a)\, b_e^r(s,a) \;+\; D \sum_{s,a} v_e(s,a)\, b_e^p(s,a) \;+\; O(D). \tag{44}$$

(Telescoping detail for (43)). Let $\mathcal{F}_k$ be the filtration up to time k and define $\xi_k := h_e^+(s_{k+1}) - \mathbb{E}[h_e^+(s_{k+1}) \mid \mathcal{F}_k]$. Then $\sum_k \xi_k$ is a martingale with bounded increments $|\xi_k| \le^(h_e^+)$, so Azuma yields an $O({}^(h_e^+)\sqrt{t_{e+1}-t_e})$ term; in UCRL2 this is absorbed into the main $\sqrt{SAN}$ terms and/or handled by the stopping rule. We keep the dominant terms explicit and absorb these standard martingale terms into $\tilde{O}$.

Step 5: Sum over epochs and count. Summing (44) over epochs $e = 1, \dots, E$:

$$Reg(N) \;\le\; \sum_{e=1}^{E} \sum_{s,a} v_e(s,a)\, b_e^r(s,a) \;+\; D \sum_{e=1}^{E} \sum_{s,a} v_e(s,a)\, b_e^p(s,a) \;+\; \tilde{O}(D\sqrt{N}), \tag{45}$$

where we used that the accumulated martingale/boundary terms are at most $\tilde{O}(D\sqrt{N})$. It remains to bound the two double sums.

Transition term. By (35), $b_e^p(s,a) \propto 1/\sqrt{\max\{1, N_e(s,a)\}}$. Using the standard UCRL2 doubling stopping rule, each visit during an epoch can be charged to an increase in $N_e(s,a)$, giving the classical counting inequality

$$\sum_{e=1}^{E} v_e(s,a)\, \frac{1}{\sqrt{\lambda + N_e(s,a)}} \;\le\; 2\sqrt{\lambda + N_N(s,a)}.$$

Summing over (s,a) and using Cauchy–Schwarz,

$$\sum_{e=1}^{E} \sum_{s,a} v_e(s,a)\, b_e^p(s,a) \;=\; \tilde{O}\left(\sum_{s,a} \sqrt{N_N(s,a)} \right) \;\le\; \tilde{O}\left(\sqrt{SAN} \right).$$

Thus the transition contribution to regret is $\tilde{O}(D\sqrt{SAN})$.

Reward term (recovery + disagreement). From (41), $b_e^r(s,a)$ is the sum of: (i) $b_e^{rec}(s,a) = \tilde{O}(\sigma\sqrt{\kappa T}/\sqrt{\eta(\lambda + N_e(s,a))})$ and (ii) $\Delta_{\max}/\sqrt{\lambda + N_e(s,a)}$. Applying the same counting inequality as above,

$$\sum_{e=1}^{E} v_e(s,a)\, b_e^{rec}(s,a) \;=\; \tilde{O}\left(\sigma \sqrt{\frac{\kappa T}{\eta}} \sum_{e=1}^{E} \frac{v_e(s,a)}{\sqrt{\lambda + N_e(s,a)}} \right) \;\le\; \tilde{O}\left(\sigma \sqrt{\frac{\kappa T}{\eta}}\, \sqrt{N_N(s,a)} \right),$$

and similarly

$$\sum_{e=1}^{E} v_e(s,a)\, \frac{\Delta_{\max}}{\sqrt{\lambda + N_e(s,a)}} \;\le\; 2\Delta_{\max}\sqrt{\lambda + N_N(s,a)}.$$

Summing over (s,a) and applying Cauchy–Schwarz yields

$$\sum_{e=1}^{E} \sum_{s,a} v_e(s,a)\, b_e^r(s,a) \;=\; \tilde{O}\left(\left(\sigma \sqrt{\tfrac{\kappa T}{\eta}} + \Delta_{\max} \right) \sqrt{SAN} \right).$$

Step 6: Conclude. Plugging the reward and transition bounds into (45) gives

$$Reg(N) \;\le\; \tilde{O}\left(\left(\sigma \sqrt{\tfrac{\kappa T}{\eta}} + \Delta_{\max} \right) \sqrt{SAN} \;+\; D\sqrt{SAN} \right),$$

and restoring the diameter factor D in the reward-to-gain conversion as in (44), we obtain (22), completing the proof. □