TRUSTMIX: REWARD LEARNING FROM AGGREGATE FEEDBACK WITH HETEROGENEOUS SURROGATES

Anonymous authors
Paper under double-blind review

ABSTRACT

When reward feedback in reinforcement learning arrives sparsely or only as a delayed aggregate, a learner may observe whether a trajectory succeeded without knowing which decisions produced the outcome. This weak credit assignment leaves local rewards poorly identified, particularly early in learning. Expert knowledge and LLM-derived reward priors can provide valuable structure, but they may be biased, with reliability varying across the state–action space. We introduce TRUSTMIX, which combines a data-driven reference model with heterogeneous structured surrogate reward models and allocates trust locally among them. Surrogates generalize to weakly observed decisions, while the reference uses observed outcomes to calibrate trust in potentially misspecified priors as evidence accumulates. We characterize when complementary surrogates improve reward estimation and provide a sublinear-regret guarantee in an idealized tabular setting. Experiments across three environments show that TRUSTMIX improves early learning while adapting its local weights as evidence accumulates.

1 INTRODUCTION

Many sequential decisions are evaluated only after an extended sequence of actions has unfolded. A treatment plan may be judged by a later clinical outcome, a recommendation policy by long-horizon user engagement, and a robot by whether it ultimately completes a task. In such problems, the learner observes the trajectory but receives only a delayed outcome or an aggregate label summarizing many decisions. We study reinforcement learning from this *aggregate feedback*, with sparse terminal rewards as an important special case.

Aggregate feedback creates two coupled difficulties. First, it provides weak temporal credit assignment: a successful trajectory does not reveal which state–action pairs caused the outcome. Second, local rewards are weakly identified when state–action pairs do not occur in sufficiently varied trajectories. Reward shaping (Ng et al., 1999), hindsight relabeling (Andrychowicz et al., 2017), intrinsic rewards (Pathak et al., 2017), and return decomposition (Arjona-Medina et al., 2019) improve exploration or credit assignment when sparse or delayed rewards are observed. Our setting is different: the learner observes only a trajectory- or segment-level label, so any local reward signal must be inferred rather than observed. The closest work studies additive trajectory feedback (Efroni et al., 2021), binary once-per-episode feedback generated by an unknown parametric model (Chatterji et al., 2021), and binary or summed segment feedback (Du et al., 2025). These approaches learn under a single feedback model; they do not address how aggregate evidence should calibrate several imperfect sources of structured reward knowledge.

Such knowledge is often available before aggregate data are sufficient. Scientific or expert models (Trella et al., 2024; Liu, 2023), rewards learned from demonstrations or preferences (Abbeel & Ng, 2004; Christiano et al., 2017), and LLM-generated reward functions (Kwon et al., 2023; Yu et al., 2023; Xie et al., 2024; Ma et al., 2024) can provide informative local structure and generalize across rarely observed decisions. However, these sources may be misspecified and reliable in different regions of the state–action space. Classical model combinations use global predictive-risk weights (van der Laan et al., 2007), whereas mixtures of experts learn input-dependent gates (Jacobs et al., 1991); related shrinkage methods trade bias against variance when combining data sources with different reliability (Rosenman et al., 2023; Chen et al., 2025). These formulations leave open the combination problem studied here: prediction errors are not locally observed, and reward models

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

fitted to the same aggregate labels have correlated estimation noise. The central question is therefore not simply whether to use prior knowledge, but where and how strongly each source should be trusted.

We introduce TRUSTMIX, a local reward-model pooling method that combines two complementary types of estimators. A tabular ridge *reference* is fitted directly to aggregate outcomes without imposing relationships across state–action pairs. It anchors the method to observed feedback but is data-hungry and uncertain in poorly identified directions. Structured *surrogates* share information across pairs and generalize earlier, but may be biased. At each state–action pair, TRUSTMIX estimates the models' joint sampling covariance and uses variance-corrected disagreement with the reference, together with a ridge-bias bound, to construct bias proxies. It then chooses convex local weights by minimizing an explicit proxy for mean squared error. Thus, surrogates can guide early learning where aggregate evidence is weak, while the reference gains influence where the local reward becomes well identified.

Our contributions are threefold. First, we formulate aggregate-feedback RL with heterogeneous sources of structured reward knowledge, including fixed expert or LLM priors and surrogates learned from the same aggregate labels. Second, we develop a covariance-aware, reference-anchored local pooling rule, characterize when complementary surrogates improve oracle local error, and establish a reference-scale sublinear-regret guarantee for an idealized tabular procedure under stated assumptions. Third, we evaluate the method on a diagnostic advertiser MDP, a substantially larger stochastic sepsis simulator, and a continuous-control study-scheduling environment calibrated with real-user memory parameters and LLM-derived priors. Across these settings, structured surrogates improve early learning, while the learned local weights adapt as aggregate evidence accumulates.

2 PROBLEM SETUP

We consider a finite tabular Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r^\star)$, where P is the transition kernel and $r^\star : \mathcal{S} \times \mathcal{A} \to [-r_{\max}, r_{\max}]$ is an unknown bounded primitive reward. Our setting includes both episodic and continuing tasks. At each primitive step, the learner observes the transition (s_t, a_t, s_{t+1}) but not $r^\star(s_t, a_t)$. Instead, feedback arrives only after a sequence of interactions, as when a treatment sequence is evaluated using a later clinical outcome or an advertising platform observes only the total revenue accumulated over a reporting period.

Let $0 = \tau_0 < \tau_1 < \cdots < \tau_n$ denote the observed feedback times, and define the ith feedback segment as $\mathcal{B}_i = \{\tau_{i-1}, \ldots, \tau_i - 1\}$. Its length $T_i = \tau_i - \tau_{i-1}$ may be fixed or random and may vary across segments. A segment may coincide with an episode, but this is not required. At time τ_i, the learner observes a single noisy aggregate label

$$y_i = \sum_{t \in \mathcal{B}_i} r^\star(s_t, a_t) + \varepsilon_i. \tag{1}$$

To express these observations in tabular form, fix an indexing map for state-action pairs

$$\kappa : \mathcal{S} \times \mathcal{A} \to \{1, \ldots, d\}, \qquad d = |\mathcal{S}||\mathcal{A}|.$$

For each segment, define the count vector $\mathbf{x}_i \in \mathbb{R}^d$ by

$$(\mathbf{x}_i)_j = \sum_{t \in \mathcal{B}_i} \mathbf{1}\{\kappa(s_t, a_t) = j\}.$$

Let $X \in \mathbb{R}^{n \times d}$ contain $\mathbf{x}_i^\top$ as its ith row, and let $\mathbf{r}^\star \in \mathbb{R}^d$ collect the primitive rewards. Stacking the observed labels and noise terms then gives

$$\mathbf{y} = X\mathbf{r}^\star + \boldsymbol{\varepsilon}. \tag{2}$$

We impose the conditional moment assumptions

$$\mathbb{E}[\boldsymbol{\varepsilon} \mid X] = \mathbf{0}, \qquad \mathrm{Cov}(\boldsymbol{\varepsilon} \mid X) = \Omega(X) \succeq 0. \tag{3}$$

Thus, the noise distribution may depend on the realized trajectories, and the aggregate labels may be heteroskedastic or correlated across feedback segments.

The additive representation in equation 1 is itself a modeling assumption. It is exact when the observed label is the sum of outcomes generated at the primitive time scale—for example, when weekly sales equal the sum of unobservable daily sales. Many tasks instead provide a non-additive terminal outcome, such as binary task success or a clinical outcome measured at discharge. In such cases, we still fit an additive model as a convenient *credit-assignment device*. The resulting per-step predictions are *pseudo-rewards*: they provide a dense learning signal whose accumulated value is intended to predict the trajectory-level outcome, but they are not interpreted as uniquely identified primitive rewards. The usefulness of a pseudo-reward is evaluated through the performance of the policy learned from it.

In this additive setting, the objective is to learn an optimal policy $\pi^\star \in \arg\max_{\pi\in\Pi} J(\pi; r^\star)$, where Π is the task-appropriate policy class and, for initial-state distribution μ_0,

$$J(\pi; r^\star) = \begin{cases} \mathbb{E}_{\mu_0,P,\pi}\left[\sum_{t=0}^{H-1} r^\star(s_t, a_t)\right], & \text{when the MDP is episodic,} \\ \liminf_{N\to\infty} \frac{1}{N}\mathbb{E}_{\mu_0,P,\pi}\left[\sum_{t=0}^{N-1} r^\star(s_t, a_t)\right], & \text{when the MDP is continuing.} \end{cases} \tag{4}$$

Here H is the episodic horizon. For the continuing average-reward case, we assume the MDP is communicating. For non-additive outcomes, the objective is to maximize the expected task outcome; the pseudo-reward serves only as an intermediate learning signal for constructing the policy.

3 METHOD

Our method begins by constructing a per-step reward signal from delayed aggregate feedback. Under the additive model, this signal estimates the primitive reward; for non-additive outcomes, it is a pseudo-reward used for credit assignment. In either case, it makes the sequence-level outcome informative about observed decisions and provides the downstream RL algorithm with immediate learning signals.

We combine two complementary types of reward models that occupy opposite sides of the bias–variance trade-off. The *reference model* uses a flexible tabular representation with no cross-pair structural restrictions on $r(s, a)$. It therefore has low structural bias and is consistent in identifiable directions under the additive model, but it is data-hungry and can have high sampling variance. The *surrogate models* impose stronger structural assumptions and encode domain knowledge from sources such as human experts or LLMs. By sharing information across state–action pairs, they can substantially reduce variance, especially early in learning, but may introduce bias when their assumptions are incorrect. We derive a statistically motivated local ensemble that combines these two sources of information at each state–action pair, then construct an exploration bonus with separate components for reward-estimation error and uncertainty in the estimated transitions.

3.1 REWARD MODELS

Reference model. We fit a tabular reference model directly to the aggregate labels. Each state–action pair has its own reward parameter, so the model imposes no functional relationship between rewards at different pairs. To obtain a unique estimate even when the count design X is rank-deficient, we use ridge regression:

$$\widetilde{\mathbf{r}} = \arg\min_{\mathbf{r}\in\mathbb{R}^d} \left\{\|\mathbf{y} - X\mathbf{r}\|_2^2 + \lambda\|\mathbf{r}\|_2^2\right\} = \left(X^\top X + \lambda I_d\right)^{-1} X^\top \mathbf{y} =: {A_\lambda}^{-1} X^\top \mathbf{y} \qquad \lambda > 0. \tag{5}$$

Ridge regularization ensures a unique solution and, when λ is small, introduces little shrinkage bias in well-identified directions; it cannot, however, recover reward components that are not identified by the data. If a state–action pair is never visited, its column in X is zero and its estimate is $\widetilde{r}_j = 0$; likewise, if two pairs have identical count columns across all feedback segments, the labels reveal only the sum of their rewards. Ridge assigns them equal estimates, but cannot determine their individual rewards.

Surrogate models. We consider M surrogate reward models $\widehat{r}_m : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. They can take three forms: (1) a low-dimensional linear reward model with domain-informed features (Ziebart et al., 2008), including fixed threshold or coarse-partition rules represented by indicator features; (2) a learned nonlinear reward model, such as a small neural network (Christiano et al., 2017), or a model whose structure is informed by domain knowledge, such as a consumer search model (Liu, 2023); and (3) an externally supplied fixed reward function, such as an expert-assigned reward table (Booth et al., 2023), an expert heuristic, or an LLM-generated reward function (Xie et al., 2024; Ma et al., 2024). In our setting, the first two types may be fitted to aggregate feedback. The third is supplied rather than fitted to those labels during a given update, although it may be revised between updates. All three encode assumptions about reward values or structure, but may be biased in different parts of the state–action space.

A surrogate may be accurate at one state–action pair but biased at another, making a single set of global weights inadequate. We therefore use local mean squared error (MSE) as the guiding principle for choosing state–action-specific convex weights, using estimated sampling covariances and proxies for bias relative to the reference model. We detail the weighting rule below.

3.2 Adaptive Local MSE-Guided Weighting

Fix a state–action pair j and collect the model predictions in $\widehat{\mathbf{r}}_j = (\widetilde{r}_j, \widehat{r}_{1,j}, \ldots, \widehat{r}_{M,j})^\top$. We seek nonnegative weights that sum to one and minimize the mixed reward's conditional mean squared error:

$$\mathbf{w}_j^{\text{oracle}} \in \arg\min_{\mathbf{w}\geq\mathbf{0},\ \mathbf{1}^\top\mathbf{w}=1} \mathbb{E}\Big[\big(\mathbf{w}^\top\widehat{\mathbf{r}}_j - r_j^\star\big)^2 \mid \mathbf{X}\Big]. \tag{6}$$

Proposition 1 (Oracle local MSE weights)**.** *Define the conditional second-moment matrix of the models' estimation errors by*

$$\mathbf{R}_j = \mathbb{E}\big[(\widehat{\mathbf{r}}_j - r_j^\star\mathbf{1})(\widehat{\mathbf{r}}_j - r_j^\star\mathbf{1})^\top \mid \mathbf{X}\big]. \tag{7}$$

For weights fixed conditional on $\mathbf{X}$*, the objective in equation 6 equals* $\mathbf{w}^\top\mathbf{R}_j\mathbf{w}$*. If* $\mathbf{R}_j$ *is positive definite and* $\mathbf{R}_j^{-1}\mathbf{1} > \mathbf{0}$ *componentwise, its unique minimizer is*

$$\mathbf{w}_j^{\text{oracle}} = \frac{\mathbf{R}_j^{-1}\mathbf{1}}{\mathbf{1}^\top\mathbf{R}_j^{-1}\mathbf{1}}. \tag{8}$$

The diagonal entries of $\mathbf{R}_j$ are the individual models' conditional mean squared errors (MSEs); each off-diagonal entry is the conditional expected product of two models' estimation errors. Equivalently, $\mathbf{R}_j = \mathbf{\Sigma}_j + \mathbf{b}_j\mathbf{b}_j^\top$, where $\mathbf{\Sigma}_j = \mathrm{Cov}(\widehat{\mathbf{r}}_j \mid \mathbf{X})$ is the covariance matrix of the reward estimates and $\mathbf{b}_j = \mathbb{E}[\widehat{\mathbf{r}}_j \mid \mathbf{X}] - r_j^\star\mathbf{1}$ is their bias vector. Because $\mathbf{R}_j$ is unknown, the following paragraphs construct a data-informed matrix estimate for choosing the weights used by the algorithm.

Covariance and bias proxies. Because reward models fitted to the same aggregate feedback can have correlated predictions, we estimate their full joint sampling covariance. This covariance is available analytically for linear or locally linear estimators and can otherwise be estimated by bootstrap; details are given in Appendix A.

Our key construction uses the data-driven reference model to estimate surrogate bias through *model disagreement*. This parallels the use of a reliable reference when combining experimental and potentially biased observational estimators (Rosenman et al., 2023; Chen et al., 2025). For surrogate m, let $\Delta_{m,j} := \widehat{r}_{m,j} - \widetilde{r}_j$. We remove the portion of squared disagreement attributable to sampling variation:

$$\begin{aligned} \widehat{v}_{\Delta,m,j} &= \widehat{\Sigma}_{j,mm} + \widehat{\Sigma}_{j,00} - 2\widehat{\Sigma}_{j,0m}, \\ \widehat{d}_{m,j} &= \sqrt{\big[\Delta_{m,j}^2 - \widehat{v}_{\Delta,m,j}\big]_+}. \end{aligned} \tag{9}$$

The ridge reference admits a coordinate-wise bias bound $|b_{0,j}| \leq B_j$. Consequently, the surrogate-bias magnitude compatible with a relative bias of size $\widehat{d}_{m,j}$ ranges from $[\widehat{d}_{m,j} - B_j]_+$ to $\widehat{d}_{m,j} + B_j$. We define

$$\widehat{b}_{0,j} = B_j, \qquad \widehat{b}_{m,j}(\beta_B) = \Big[\widehat{d}_{m,j} + \beta_B B_j\Big]_+, \qquad \beta_B \in [-1, 1]. \tag{10}$$

Thus, $\beta_B = -1$ selects the surrogate-favoring lower endpoint, whereas $\beta_B = +1$ selects the reference-favoring upper endpoint; intermediate values encode the desired level of trust in the surrogate. The derivation and limitations of these proxies are given in Appendix B.

Local weight construction. Let $\widehat{\mathbf{b}}_j = (\widehat{b}_{0,j}, \ldots, \widehat{b}_{M,j})^\top$ and define $\widehat{\mathbf{R}}_j = \widehat{\mathbf{\Sigma}}_j + \widehat{\mathbf{b}}_j \widehat{\mathbf{b}}_j^\top$. The implemented local weights are

$$\widehat{\mathbf{w}}_j \in \arg\min_{\mathbf{w} \geq \mathbf{0},\ \mathbf{1}^\top \mathbf{w} = 1} \mathbf{w}^\top \widehat{\mathbf{R}}_j \mathbf{w}. \tag{11}$$

Mixed reward and exploration bonuses. For a state–action pair indexed by j, the mixed reward estimate is $\widehat{r}_{\mathrm{mix},j} = \widehat{\mathbf{w}}_j^\top \widehat{\mathbf{r}}_j$. We define its reward-model exploration bonus as $b_{r,j} = \sqrt{\widehat{\mathbf{w}}_j^\top \widehat{\mathbf{R}}_j \widehat{\mathbf{w}}_j}$. Thus, $b_{r,j}^2$ is the optimized local MSE proxy for the mixed reward: it is large where the reward models still provide an imprecise or potentially biased estimate and decreases as their estimated local error decreases. We separately encourage coverage of the transition dynamics using the count-based bonus $b_P(s,a) = 1/\sqrt{N(s,a)+1}$, where $N(s,a)$ is the number of observed transitions from (s,a). The data-collection policy is obtained by planning in the estimated MDP with the augmented reward

$$r_{\mathrm{plan}}(s,a) = \widehat{r}_{\mathrm{mix},j(s,a)} + \beta_r b_{r,j(s,a)} + \beta_P b_P(s,a), \qquad \beta_r, \beta_P \geq 0. \tag{12}$$

This construction is inspired by optimism-based exploration in bandits and reinforcement learning (Auer et al., 2002; Strehl & Littman, 2008). In particular, $b_P(s,a)$ has the classical $1/\sqrt{N(s,a)}$ dependence used in count-based exploration, whereas $b_{r,j}$ is specific to our setting and reflects the estimated local error of the mixed reward. The coefficients β_r and β_P control the contributions of the two exploration incentives.

Policy learning and extension to continuous spaces. Beginning with an initial set of trajectories and aggregate labels, we repeatedly fit the reward models, estimate their covariance and bias proxies, and construct the local mixture weights. In the tabular setting, we combine the estimated transition model with r_{plan} to obtain the next data-collection policy. The new trajectories and labels are added to the dataset before the models and policy are updated again. After data collection, we refit all models and compute the final policy using only $\widehat{r}_{\mathrm{mix}}$, equivalently setting $\beta_r = \beta_P = 0$.

We also provide a heuristic extension to continuous state–action spaces. In this case, we replace the ridge reference with an expressive DNN trained by matching each aggregate label to the sum of its per-step predictions. The DNN provides a flexible, relatively low-bias but high-variance reference, while the surrogate models impose stronger structure to reduce variance. We evaluate the local mixture on replay-buffer samples and use the mixed reward and reward-model bonus to relabel experience for the soft actor-critic (SAC) algorithm. Because exact state–action counts are uninformative in continuous spaces, we set $\beta_P = 0$ and rely on SAC's entropy-based exploration. Complete procedures are provided in Algorithms 1 and 2 in the appendix.

4 Theoretical Analysis

We ask two questions: when do surrogates reduce the local error of the reference, and can mixing them in ever cost the no-regret rate of a reference-only learner? Proposition 2 answers the first for the oracle weights of Proposition 1; Theorem 1 answers the second for the implemented weights. All proofs are in Appendix C.

Proposition 2 (Benefit of complementary surrogates)**.** *Fix a pair j and suppose the M surrogates satisfy $|b_{m,j}| \leq \bar{b}$, $\Sigma_{j,mm} \leq \sigma_s^2$, and $\Sigma_{j,m\ell} \leq \rho\sqrt{\Sigma_{j,mm}\Sigma_{j,\ell\ell}}$ for $m \neq \ell$, where $\bar{b}, \sigma_s \geq 0$ and $0 \leq \rho < 1$. Then*

$$\mathbb{E}\Big[\big((\mathbf{w}_j^{\mathrm{oracle}})^\top \widehat{\mathbf{r}}_j - r_j^\star\big)^2 \mid \mathbf{X}\Big] \leq \min\Big\{R_{j,00},\ \bar{b}^2 + \sigma_s^2\big(\rho + \tfrac{1-\rho}{M}\big)\Big\},$$

so the oracle mixture strictly improves on the reference whenever the second term is smaller than $R_{j,00}$.

Interpretation. Surrogates never hurt the oracle, which can always fall back on the reference, and they help when they are individually not too biased and mutually complementary: M enters only

through $(1-\rho)/M$, so near-duplicate surrogates add nothing. The gain is largest where the reference is weakest (at an unvisited pair $R_{j,00} = (r_j^\star)^2$) and is expected to fade as the reference's confidence radius shrinks, which is exactly when the local weights in equation 11 should shift back to the reference.

Regret of the collection policy. We analyze an idealized version of Algorithm 1: $S = |\mathcal{S}|$, $A = |\mathcal{A}|$, $d = SA$, stationary P, $|r^\star(s,a)| \le r_{\max}$, and initial distribution μ_0. After an initial dataset of at most N_0 steps, in collection round k one deterministic stationary mapping $\pi_k \in \Pi_{\mathrm{stat}}$, the class of all such mappings, is run for a block of $H_k \le H$ steps from μ_0; the block is partitioned into feedback segments fixed in advance, all labels arrive before the next update, and models and weights are frozen within the block. The planner uses the empirical transition frequencies $\widehat{P}_k$, fixed-feature ridge surrogates or fixed external tables, the clustered sandwich covariance, and the bias proxies equation 10 with any fixed $\beta_B \in [-1,1]$. Let $N = \sum_k H_k$, $t_k = \sum_{k' \le k} H_{k'}$, and let $N_k(j)$ count visits to pair j before block k. Appendix C.2 gives the full specification and the confidence constants $C_N, \ell_N, c_{P,N}$, which grow at most logarithmically in the budget.

Assumption 1 (Sequential aggregate noise)**.** *For every trajectory g, $y_g = X_g \mathbf{r}^\star + \varepsilon_g$ and, almost surely,*

$$\mathbb{E}[\exp(v^\top \varepsilon_g) \mid \mathcal{G}_g] \le \exp(\sigma^2 \|v\|_2^2/2) \quad \textit{for all } v, \tag{13}$$

where $\mathcal{G}_g$ contains all past data and the current trajectory before its labels are revealed.

Assumption 2 (Uniform probability of holding)**.** *$P(s \mid s,a) \ge p$ for a fixed $p \in (0,1)$ and every pair that may be selected.*

Assumption 3 (Approximate stationary planning)**.** *Let $J_h(\pi; r, P)$ be the expected h-step return from μ_0 and $\widehat{J}_k(\pi) = J_{H_k}(\pi; r_{\mathrm{plan},k}, \widehat{P}_k)$. The planner returns $\pi_k \in \Pi_{\mathrm{stat}}$ with $\widehat{J}_k(\pi_k) \ge \max_{\pi \in \Pi_{\mathrm{stat}}} \widehat{J}_k(\pi) - c_{\mathrm{opt}} H_k/\sqrt{t_k}$ for a constant $c_{\mathrm{opt}} \ge 0$.*

Regret is measured against the best stationary mapping at each horizon,

$$\mathrm{Reg}_{\mathrm{stat}}(K) := \sum_{k=1}^{K} \Big[\max_{\pi \in \Pi_{\mathrm{stat}}} J_{H_k}(\pi; r^\star, P) - J_{H_k}(\pi_k; r^\star, P) \Big]. \tag{14}$$

Theorem 1 (Regret with adaptive mixing)**.** *Under Assumptions 1–3 and the specification of Appendix C.2, fix $\delta \in (0,1)$ and set $\beta_r = 1$ and $\beta_P = 4C_N + H r_{\max} c_{P,N}$ (any constant multiple $c \ge 1$ of this β_P is admissible). For any fixed $\beta_B \in [-1,1]$, with probability at least $1-\delta$,*

$$\mathrm{Reg}_{\mathrm{stat}}(K) \le \widetilde{O}\Big((C_N + H r_{\max}\sqrt{S}) \big[\sqrt{SAN} + H^3 S^2 A \big] \Big) + 2 c_{\mathrm{opt}} \sqrt{N}. \tag{15}$$

Interpretation. The theorem is a safety guarantee rather than a speed-up claim: for fixed S, A, H, p, σ, N_0, fixed ridge penalties, and fixed surrogate features, it gives the $\widetilde{O}(\sqrt{N})$ rate of a reference-only learner (equivalently, $\widetilde{O}(\sqrt{K})$ for uniformly bounded, nonzero block lengths), and the surrogates enter only through the constant $Q \ge 1$ in C_N, so arbitrarily biased surrogates cannot induce linear regret. The reason is Lemma 2: since putting all weight on the reference is feasible in equation 11, the mixed reward stays within $b_{r,j} + 4q_{k,j}$ of $r_j^\star$ (block index suppressed), where $q_{k,j} = C_N/\sqrt{N_k(j)+1}$ is the reference's confidence radius, so the bonuses keep planning optimistic. Together with Proposition 2, this shows that surrogates can lower local error where the reference is weak, while the mixing rule remains controlled at the reference scale. The result covers the collection policies of the idealized procedure, not the final bonus-free policy (Appendix C.14).

5 Experiments and Results

5.1 Controlled Misspecification in the Advertiser MDP

Setting. We first test whether TrustMix can exploit useful but misspecified reward models when aggregate feedback is scarce, while eventually shifting toward a more flexible reference model. We use a continuing tabular MDP with 10 shopper-traffic states and three advertising actions: low, medium, and high intensity. Advertising affects the stochastic evolution of traffic, and the environment does not reset between feedback blocks. The latent daily profit combines a concave benefit

from traffic, a convex advertising cost, and a state–action interaction, making the optimal action depend on the current traffic level.

The learner observes every transition but not the daily profit. After each 30-day block, it receives only the noisy aggregate profit for that block. TrustMix combines a 30-parameter tabular reference model with two lower-dimensional models fitted to the same aggregate labels: a bilinear surrogate that omits the quadratic terms in the true reward, and a threshold surrogate that assigns action-specific rewards separately to low- and high-traffic states. Both surrogates are therefore structurally informative but deliberately misspecified. We compare against delayed-feedback Differential Q-learning, aggregate REINFORCE, reference-only planning, and a Uniform Weight ablation that fixes the three reward-model weights to $1/3$. All methods share the same ten-block uniform-policy initialization and are evaluated over 50 matched seeds. Full environment and algorithm specifications are given in Appendix E.1.

Results. TrustMix achieves the lowest sample-mean policy gap at every reported budget and the lowest cumulative regret (Table 1 and Figure 1(a)). Its advantage is largest early in learning: at 50 blocks, its scaled policy gap is 23.40, compared with 52.85 for reference-only. By 100 blocks, reference-only has largely caught up (17.37 versus 16.75), showing that the main benefit of the surrogates is improved early data efficiency. Consistently, at 9,900 steps TrustMix has mean cumulative regret 225.62, below Uniform Weight (263.29) and reference-only (286.13).

The weight evolution confirms the intended bias–variance tradeoff (Figure 1(b)). After ten blocks, TrustMix assigns weight 0.824 to the low-dimensional, lower-variance bilinear surrogate and almost none to the data-hungry tabular reference. As more aggregate feedback arrives, the reference weight rises to 0.417 after 50 blocks and 0.845 after 1,000 blocks. TrustMix therefore exploits structured surrogates when data are scarce and progressively shifts toward the more expressive data-driven reference model as its estimation variance decreases.

5.2 Sparse Terminal Feedback in a Sepsis Simulator

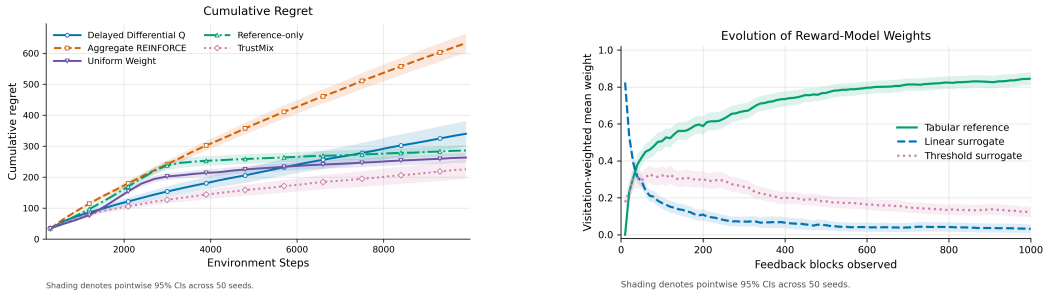
Setting. We next evaluate TrustMix in a substantially larger episodic MDP with sparse terminal feedback. We adapt the synthetic sepsis simulator of Oberst and Sontag (Oberst & Sontag, 2019). A state records diabetes status, four discretized physiological measurements, and whether antibiotics, vasopressors, and mechanical ventilation are active. Including separate absorbing death and discharge states gives $1{,}442$ states. The eight actions represent all binary combinations of the three treatments, and each episode contains at most $H = 20$ treatment decisions.

We retain the simulator's hand-specified physiological transitions but replace its deterministic demise rule with a stochastic benchmark in which mortality risk increases with the number and type of abnormal physiological indicators. The reward is $+1$ for discharge, -1 for death, and zero otherwise. The agent observes the trajectory but receives only the aggregate terminal outcome after the episode; a truncated episode receives label zero.

TrustMix combines an $11{,}536$-parameter tabular reference with two learned surrogates fitted to the same episode labels: a 112-parameter physiology main-effects model and an 80-parameter severity-group model. It additionally uses a fixed doctor-prior score that rewards selecting no treatment in fully normal states and penalizes states with at least three abnormal indicators. We compare against delayed Q-learning, aggregate REINFORCE, the trajectory-feedback UCBVI-TS adaptation (Efroni et al., 2021), reference-only planning, and TrustMix without the doctor prior. All methods share ten uniform-policy warm-up episodes and are evaluated for 600 episodes over 50 matched seeds. Complete details appear in Appendix E.2.

Results. Both TrustMix variants substantially outperform the standard baselines (Table 2 and Figure 2(a)). Full TrustMix is especially effective in the low-data regime: at 50 episodes, its policy gap is 0.641, compared with 0.770 without the doctor prior, 0.802 for UCBVI-TS, and 0.841 for reference-only. It also achieves the lowest cumulative pseudo-regret over 600 episodes (381.03, versus 397.16 without the prior and 487.31–509.62 for the other methods).

Comparing the two TrustMix variants isolates the benefit of domain knowledge. The doctor prior provides a clear early advantage, whereas the no-prior variant eventually attains a lower final gap at 600 episodes (0.532 versus 0.580) as more data become available. Consistently, the doctor-prior



(a) Cumulative regret.

(b) TrustMix reward-model weights.

Figure 1: **Advertiser MDP.** Left: cumulative regret over the declared early learning window of 9,900 environment steps. Right: visitation-weighted mean TrustMix weights over a longer run. Curves show means and shaded regions show pointwise 95% normal confidence intervals across 50 matched seeds.

Table 1: Policy optimality gap at fixed data budgets. One feedback block represents one month and contains 30 environment steps. Entries report $1000 \times [g(\pi^\star) - g(\widehat{\pi})]$ as the mean $\pm$ pointwise 95% normal-CI half-width (1.96 SE) across 50 matched seeds. Lower is better.

Blocks	Delayed Q	REINFORCE	Reference-only	TrustMix
10	127.52 ± 27.76	92.86 ± 4.21	74.90 ± 16.08	$\mathbf{55.01 \pm 11.83}$
20	74.66 ± 19.44	84.44 ± 4.80	61.33 ± 18.11	$\mathbf{38.64 \pm 8.72}$
50	61.56 ± 17.47	74.29 ± 4.30	52.85 ± 15.86	$\mathbf{23.40 \pm 4.66}$
100	53.08 ± 17.16	67.99 ± 4.21	17.37 ± 10.41	$\mathbf{16.75 \pm 5.39}$

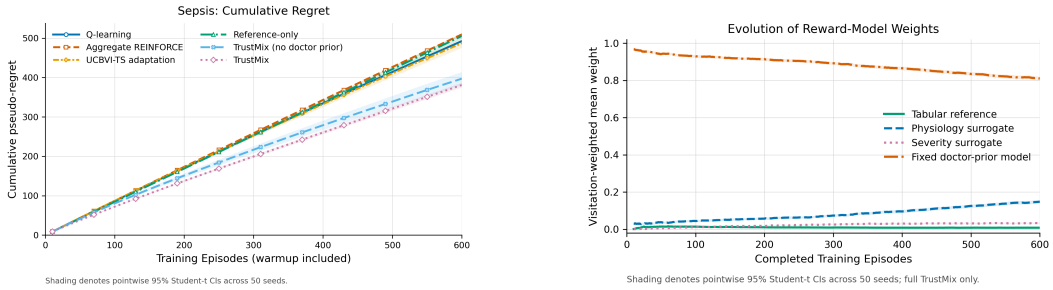
Here $g(\pi) := \liminf_{N\to\infty} N^{-1}\mathbb{E}_{\mu_0,P,\pi}[\sum_{t=0}^{N-1} r^\star(s_t, a_t)]$. Bold denotes the lowest sample mean among methods shown, not statistical significance.

weight decreases from 0.968 to 0.811, while the physiology surrogate grows from 0.031 to 0.147 (Figure 2(b)). Thus, an informative prior accelerates early learning despite its misspecification, while learned reward models become increasingly useful as evidence accumulates.

5.3 Real-World-Calibrated Continuous Control in FSRS-Exam

Setting. Finally, we evaluate LLM-derived reward priors in FSRS-Exam, a continuous-control simulator calibrated using real-user data. A student studies one of ten topics per day for 20 days and takes a ten-question exam on day 21. The state records the day and each topic's review count, lapses, elapsed time, difficulty, and memory stability. A continuous action assigns logits to the ten topics, and the reviewed topic is sampled from the resulting softmax distribution. No daily reward is observed; the agent receives only the average final exam score. State transitions follow FSRS-6, an open-source memory model used by Anki (Ye et al., 2022; Su et al., 2023; Open Spaced Repetition, 2026b; Anki, 2026). Each evaluation seed uses one of 50 held-out real-user profiles fitted to real Anki review histories (Open Spaced Repetition, 2026c;a), while four disjoint users are reserved for model and hyperparameter selection. Thus, although the study protocol is designed, the memory dynamics are calibrated from real behavior. Claude Opus, Claude Sonnet, GPT-5.6, and GPT-6 rank which topic should be studied in sampled learner states without access to the FSRS equations, user parameters, or exam outcomes. Their responses are used to train four fixed surrogate reward models. TrustMix combines these surrogates with a neural reference model to train SAC (Haarnoja et al., 2018). We compare against delayed-reward SAC, RUDDER-SAC (Arjona-Medina et al., 2019), and reference-only SAC, with uniform and round-robin policies as heuristic controls. Results use 50 matched held-out profiles and 500 online episodes; full details are given in Appendix E.3.

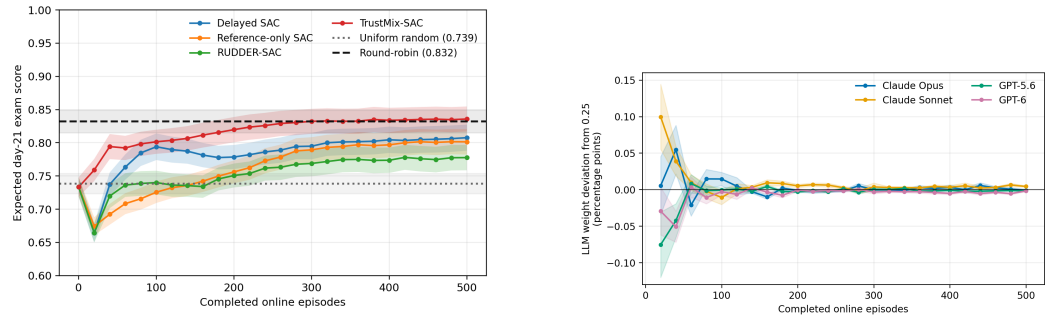
Results. TrustMix-SAC learns faster than all learned baselines and is the only learned method to reach the performance of the strong round-robin heuristic, doing so after approximately 300 episodes (Figure 3(a)). Its normalized learning-curve AUC is 0.8204, compared with 0.7878 for delayed SAC, 0.7654 for reference-only SAC, and 0.7562 for RUDDER-SAC. After 500 episodes, TrustMix attains an expected exam score of 0.8358, slightly above round-robin at 0.8322. Full numerical



(a) Cumulative pseudo-regret. (b) TrustMix reward-model weights.

Figure 2: **Sepsis simulator.** Left: cumulative pseudo-regret of the collection policies. Right: visitation-weighted mean reward-model weights for full TrustMix. Curves show means and shaded regions show pointwise 95% Student-t confidence intervals across 50 matched seeds.

comparisons are reported in Table 3 in the appendix. TrustMix assigns approximately equal weight to the four LLM surrogates after an initial transient (Figure 3(b)). Although their learned reward functions are correlated, they are not identical (Figure 4 in the appendix). Thus, the improvement does not come from selecting one dominant model, but from combining multiple nonidentical LLM priors about learning and memory. The result shows that such priors can provide useful process-level guidance when only a sparse final outcome is observed.



(a) Expected day-21 exam score. **(b)** LLM weight deviations from 0.25.

Figure 3: **FSRS-EXAM.** Left: expected exam score of the current policy over 500 online episodes. Right: deviations of the four LLM mixture weights from equal weighting. Lines and shaded regions show means $\pm$ one standard error across 50 paired held-out user profiles. Horizontal lines in panel (a) show the uniform-random and round-robin heuristics.

6 LIMITATIONS

TrustMix benefits only when at least one surrogate provides useful structure. Reference disagreement is an imperfect bias proxy, particularly before the reference has sufficient coverage. A low-variance surrogate may improve early learning yet become limiting as its misspecification dominates or its guidance reduces exploration; therefore, early gains need not imply permanent dominance over model-free methods. Finally, our regret guarantee applies to the tabular setting, while the neural SAC extension is validated empirically.

7 CONCLUSION

We introduced TrustMix, a reference-anchored method for locally combining heterogeneous reward surrogates using covariance and bias–variance information. Our analysis establishes when surrogates reduce estimation error while preserving the reference learner's sublinear regret rate. Across three environments, including real-world-calibrated FSRS-EXAM, TrustMix improves early learning and cumulative performance. It is therefore especially useful when interaction is expensive or informative rewards are sparse and delayed, making early data efficiency essential.

AI USE STATEMENT

Generative AI tools were used in two roles. First, as part of the proposed method, four language models produced topic-priority rankings that were used to train the frozen surrogate reward models in FSRS-EXAM. Second, AI assistants supported literature discovery, experimental-design feedback, code implementation and debugging, result analysis and visualization, and editing the manuscript for clarity.

The research question, core methodology, and theoretical results were developed by the authors. All AI-assisted code was manually reviewed and validated through unit tests, controlled experiments, and reproducibility checks. Citations and literature summaries were checked against the original sources, numerical claims were verified from saved experimental outputs, and the authors independently reviewed all mathematical statements and proofs. The authors take full responsibility for the paper and all AI-assisted content.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, 2011. URL https://proceedings.neurips.cc/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html.

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004. doi: 10.1145/1015330.1015430. URL https://ai.stanford.edu/~ang/papers/icml04-apprentice.pdf.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Anki. Deck options: Free spaced repetition scheduler. https://docs.ankiweb.net/deck-options.html#fsrs, 2026. Accessed September 2026.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. doi: 10.1023/A:1013689704352.

Serena Booth, W. Bradley Knox, Julie Shah, Scott Niekum, Peter Stone, and Alessandro Allievi. The perils of trial-and-error reward design: Misdesign through overfitting and invalid task specifications. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5920–5929, 2023. doi: 10.1609/aaai.v37i5.25733.

Niladri S. Chatterji, Aldo Pacchiano, Peter L. Bartlett, and Michael I. Jordan. On the theory of reinforcement learning with once-per-episode feedback. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/1bf2efbbe0c49b9f567c2e40f645279a-Abstract.html.

Shuxiao Chen, Sai Li, Bo Zhang, and Ting Ye. Minimax rates and adaptivity in combining experimental and observational data. *Journal of Causal Inference*, 13(1):82–93, 2025. doi: 10.1515/jci-2024-0024.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Yihan Du, Anna Winnicki, Gal Dalal, Shie Mannor, and R. Srikant. Reinforcement learning with segment feedback. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14598–14647. PMLR, 2025. URL https://proceedings.mlr.press/v267/du25e.html.

Yonathan Efroni, Nadav Merlis, and Shie Mannor. Reinforcement learning with trajectory feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7288–7295, 2021. doi: 10.1609/aaai.v35i8.16895. URL https://doi.org/10.1609/aaai.v35i8.16895.

C. A. Field and A. H. Welsh. Bootstrapping clustered data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):369–390, 2007. doi: 10.1111/j.1467-9868.2007.00593.x.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, 2018.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=10uNUgI5Kl.

Kung-Yee Liang and Scott L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986. doi: 10.1093/biomet/73.1.13.

Xiao Liu. Dynamic coupon targeting using batch deep reinforcement learning: An application to livestream shopping. *Marketing Science*, 42(4):637–658, 2023. doi: 10.1287/mksc.2022.1403.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70c26937fbf3d4600b69a129031b66ec-Abstract-Conference.html.

Whitney K. Newey and Daniel McFadden. Large sample estimation and hypothesis testing. In Robert F. Engle and Daniel L. McFadden (eds.), *Handbook of Econometrics*, volume 4, pp. 2111–2245. Elsevier, 1994. doi: 10.1016/S1573-4412(05)80005-4.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999. URL https://ai.stanford.edu/~ang/papers/shaping-icml99.pdf.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with Gumbel-max structural causal models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4881–4890. PMLR, 2019. URL https://proceedings.mlr.press/v97/oberst19a.html.

Open Spaced Repetition. Anki-button-usage: Button usage in the Anki review dataset. https://github.com/open-spaced-repetition/Anki-button-usage, 2026a. Commit 585d20d. Accessed September 2026.

Open Spaced Repetition. py-fsrs: Python package for FSRS spaced repetition. https://github.com/open-spaced-repetition/py-fsrs, 2026b. Version 6.3.2, commit 9446cb0. Accessed September 2026.

Open Spaced Repetition. srs-benchmark: A benchmark for spaced repetition schedulers/algorithms. https://github.com/open-spaced-repetition/srs-benchmark, 2026c. Commit bd9110f. Accessed September 2026.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 2017. URL https://proceedings.mlr.press/v70/pathak17a.html.

Evan T. R. Rosenman, Guillaume Basse, Art B. Owen, and Mike Baiocchi. Combining observational and experimental datasets using shrinkage estimators. *Biometrics*, 79(4):2961–2973, 2023. doi: 10.1111/biom.13827.

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. doi: 10.1016/j.jcss.2007.08.009.

Jingyong Su, Junyao Ye, Liqiang Nie, Yilong Cao, and Yongyong Chen. Optimizing spaced repetition schedule by capturing the dynamics of memory. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10085–10097, 2023. doi: 10.1109/TKDE.2023.3251721.

Anna L. Trella, Kelly W. Zhang, Inbal Nahum-Shani, Vivek Shetty, Finale Doshi-Velez, and Susan A. Murphy. Reward design for an online reinforcement learning algorithm supporting oral self-care. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(13):15724–15730, 2024. doi: 10.1609/aaai.v37i13.26866.

Joel A. Tropp. User-friendly tail bounds for matrix martingales. Technical report, California Institute of Technology, 2011. URL https://authors.library.caltech.edu/records/stacd-2cs60.

Mark J. van der Laan, Eric C. Polley, and Alan E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1):Article 25, 2007. doi: 10.2202/1544-6115.1309.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3–4):279–292, 1992. doi: 10.1007/BF00992698. URL https://doi.org/10.1007/BF00992698.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9941833e8327910ef25daeb9005e4748-Abstract-Conference.html.

Junyao Ye, Jingyong Su, and Yilong Cao. A stochastic shortest path algorithm for optimizing spaced repetition scheduling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4381–4390. ACM, 2022. doi: 10.1145/3534678.3539081.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 374–404. PMLR, 2023. URL https://proceedings.mlr.press/v229/yu23a.html.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008.

A Estimation of the Joint Prediction Covariance

Index the reference model by $m = 0$ and the surrogates by $m = 1, \dots, M$. Let $g = 1, \dots, G$ index complete trajectories, and let $\mathcal{I}_g$ contain the feedback segments belonging to trajectory g. We use the working approximation that trajectories are independent clusters conditional on the policy used in each collection round,[1] while allowing arbitrary heteroskedasticity and dependence among feedback segments within the same trajectory.

Linear reward models. For a fixed-feature linear model m, let $\boldsymbol{\phi}_m(s, a) \in \mathbb{R}^{p_m}$ denote its feature vector. For feedback segment $\mathcal{B}_i$, define the aggregated feature vector

$$\mathbf{z}_{m,i} = \sum_{t \in \mathcal{B}_i} \boldsymbol{\phi}_m(s_t, a_t),$$

and let $\mathbf{Z}_m$ contain $\mathbf{z}_{m,i}^\top$ as its rows. The fitted parameter and the prediction at state–action pair j are

$$\widehat{\boldsymbol{\theta}}_m = \mathbf{A}_m^{-1} \mathbf{Z}_m^\top \mathbf{y}, \qquad \widehat{r}_{m,j} = \boldsymbol{\phi}_m(j)^\top \widehat{\boldsymbol{\theta}}_m, \qquad \mathbf{A}_m = \mathbf{Z}_m^\top \mathbf{Z}_m + \lambda_m I.$$

The tabular reference is the special case $\mathbf{Z}_0 = \mathbf{X}$, $\boldsymbol{\phi}_0(j) = e_j$, and $\lambda_0 = \lambda$.

Define

$$\mathbf{a}_{m,j} = \mathbf{Z}_m \mathbf{A}_m^{-1} \boldsymbol{\phi}_m(j),$$

so that $\widehat{r}_{m,j} = \mathbf{a}_{m,j}^\top \mathbf{y}$. Consequently, for two linear models m and ℓ,

$$\mathrm{Cov}\left(\widehat{r}_{m,j}, \widehat{r}_{\ell,j} \mid \{\mathbf{Z}_q\}_{q=0}^M\right) = \mathbf{a}_{m,j}^\top \boldsymbol{\Omega} \mathbf{a}_{\ell,j}, \tag{16}$$

where $\boldsymbol{\Omega} = \mathrm{Cov}(\boldsymbol{\varepsilon} \mid \{\mathbf{Z}_q\})$. Under trajectory clustering, $\boldsymbol{\Omega} = \mathrm{blockdiag}(\boldsymbol{\Omega}_1, \dots, \boldsymbol{\Omega}_G)$, so

$$\mathrm{Cov}\left(\widehat{r}_{m,j}, \widehat{r}_{\ell,j} \mid \{\mathbf{Z}_q\}_{q=0}^M\right) = \sum_{g=1}^G \mathbf{a}_{m,j,g}^\top \boldsymbol{\Omega}_g \mathbf{a}_{\ell,j,g},$$

where $\mathbf{a}_{m,j,g}$ contains the entries associated with trajectory g.

We estimate this quantity with a clustered residual sandwich estimator (Liang & Zeger, 1986). Let $\widehat{\boldsymbol{\varepsilon}}_g$ denote the reference-model residuals for trajectory g, and define the cluster contribution

$$\widehat{q}_{m,j,g} = \mathbf{a}_{m,j,g}^\top \widehat{\boldsymbol{\varepsilon}}_g.$$

After centering these contributions across trajectories, $\widetilde{q}_{m,j,g} = \widehat{q}_{m,j,g} - G^{-1} \sum_{h=1}^G \widehat{q}_{m,j,h}$, the estimated cross-model prediction covariance is

$$\widehat{\Sigma}_{j,m\ell} = \sum_{g=1}^G \widetilde{q}_{m,j,g} \widetilde{q}_{\ell,j,g}. \tag{17}$$

The diagonal case $m = \ell$ gives the usual clustered prediction variance. The off-diagonal entries use products of the two models' contributions from the same trajectory and therefore retain their shared-label dependence. A fixed external surrogate has no label-dependent contribution and is assigned zero sampling variance and covariance.

Nonlinear reward models with asymptotically linear estimators. For a regular low-dimensional nonlinear model, suppose $\widehat{\boldsymbol{\theta}}_m$ is characterized by trajectory-level estimating equations

$$\sum_{g=1}^G \boldsymbol{\psi}_{m,g}(\widehat{\boldsymbol{\theta}}_m) = \mathbf{0}.$$

[1] Under adaptive data collection, trajectories from different rounds are not unconditionally independent: earlier trajectories are used to construct the policy that generates later trajectories. Conditional on the realized policy, independently initialized rollouts within a collection round may nevertheless be treated as approximately independent. Our clustered covariance estimator conditions on the realized sequence of policies and does not capture uncertainty arising from the policy updates themselves.

Let

$$\widehat{\mathbf{H}}_m = \sum_{g=1}^{G} \frac{\partial \boldsymbol{\psi}_{m,g}(\widehat{\boldsymbol{\theta}}_m)}{\partial \boldsymbol{\theta}_m^\top}.$$

The estimated contribution of trajectory g to the parameter's influence function is

$$\widehat{\boldsymbol{\xi}}_{m,g} = -\widehat{\mathbf{H}}_m^{-1} \boldsymbol{\psi}_{m,g}(\widehat{\boldsymbol{\theta}}_m).$$

The cluster-robust parameter cross-covariance between models m and ℓ is therefore

$$\widehat{\mathrm{Cov}}\left(\widehat{\boldsymbol{\theta}}_m, \widehat{\boldsymbol{\theta}}_\ell\right) = \sum_{g=1}^{G} \widetilde{\boldsymbol{\xi}}_{m,g} \widetilde{\boldsymbol{\xi}}_{\ell,g}^\top, \tag{18}$$

where the influence contributions are centered across trajectories. The off-diagonal expression in Equation equation 18 is essential when the models use the same labels.

For maximum-likelihood estimation, $\boldsymbol{\psi}_{m,g}$ is the score contributed by trajectory g, and $\widehat{\mathbf{H}}_m$ is the corresponding observed score derivative. For a generalized method-of-moments estimator with cluster moment $\mathbf{g}_{m,g}(\boldsymbol{\theta}_m)$, moment Jacobian $\widehat{\mathbf{D}}_m$, and weighting matrix $\widehat{\mathbf{W}}_m$, the influence contribution can equivalently be written as

$$\widehat{\boldsymbol{\xi}}_{m,g} = -\frac{1}{G}\left(\widehat{\mathbf{D}}_m^\top \widehat{\mathbf{W}}_m \widehat{\mathbf{D}}_m\right)^{-1} \widehat{\mathbf{D}}_m^\top \widehat{\mathbf{W}}_m \mathbf{g}_{m,g}(\widehat{\boldsymbol{\theta}}_m)$$

(Newey & McFadden, 1994).

For a nonlinear reward prediction $\widehat{r}_{m,j} = r_m(j; \widehat{\boldsymbol{\theta}}_m)$, define its parameter gradient

$$\widehat{\mathbf{d}}_{m,j} = \nabla_{\boldsymbol{\theta}_m} r_m(j; \widehat{\boldsymbol{\theta}}_m).$$

The delta method then gives

$$\widehat{\Sigma}_{j,m\ell} = \widehat{\mathbf{d}}_{m,j}^\top \widehat{\mathrm{Cov}}\left(\widehat{\boldsymbol{\theta}}_m, \widehat{\boldsymbol{\theta}}_\ell\right) \widehat{\mathbf{d}}_{\ell,j}. \tag{19}$$

Equivalently, define the scalar prediction influence $\widehat{q}_{m,j,g} = \widehat{\mathbf{d}}_{m,j}^\top \widehat{\boldsymbol{\xi}}_{m,g}$ and apply Equation equation 17. This representation also provides cross-covariances between different estimator types: for example, the covariance between a ridge prediction and a maximum-likelihood prediction is obtained by multiplying their trajectory-level prediction influences for the same trajectory. Estimating only the models' separate standard errors would omit these cross-model terms.

Paired trajectory bootstrap. When a reliable influence-function calculation is unavailable, we use a paired trajectory bootstrap (Field & Welsh, 2007). For each replicate b, we resample complete trajectories with replacement and refit every data-dependent reward model using the same resampled trajectories. A stratified variant can resample separately within each collection round, preserving the number collected under each policy. Externally supplied surrogates remain fixed. In extscFSRS-Exam, the only data-dependent reward model is the reference, and episodes are resampled from the pooled set of completed episodes.

Let

$$\widehat{\mathbf{r}}_j^{(b)} = \left(\widehat{r}_{0,j}^{(b)}, \ldots, \widehat{r}_{M,j}^{(b)}\right)^\top$$

collect the predictions from bootstrap replicate b, and let $\overline{\mathbf{r}}_j^{\mathrm{boot}}$ be their mean across replicates. The joint covariance estimate is

$$\widehat{\boldsymbol{\Sigma}}_j^{\mathrm{boot}} = \frac{1}{B_{\mathrm{boot}} - 1} \sum_{b=1}^{B_{\mathrm{boot}}} \left(\widehat{\mathbf{r}}_j^{(b)} - \overline{\mathbf{r}}_j^{\mathrm{boot}}\right)\left(\widehat{\mathbf{r}}_j^{(b)} - \overline{\mathbf{r}}_j^{\mathrm{boot}}\right)^\top. \tag{20}$$

Using the same resample for every model is what identifies the off-diagonal covariance terms. If any included model requires the bootstrap, we use the paired bootstrap for the entire joint prediction vector rather than combining separately estimated covariance blocks.

This procedure resamples both the labels and their trajectory designs, so it approximates repeated-sample prediction covariance rather than the covariance conditional on the observed design. Furthermore, the bootstrap conditions on the realized data-collection process and does not reproduce uncertainty from the policy updates themselves. In the continuous extension, each bootstrap member is a separately initialized full DNN trained on a resample of complete episodes. Thus, its predictive variation includes both episode-resampling variation and variation from neural-network initialization and optimization; it is not a frozen-feature or last-layer bootstrap.

B Derivation of the Bias Proxies

This section derives the bias proxies in Equation equation 10. Let

$$A_\lambda := \mathbf{X}^\top \mathbf{X} + \lambda I_d, \qquad \widetilde{\mathbf{r}} := A_\lambda^{-1} \mathbf{X}^\top \mathbf{y}$$

denote the ridge reference estimator. Under the aggregate-feedback model

$$\mathbf{y} = \mathbf{X}\mathbf{r}^\star + \boldsymbol{\varepsilon}, \qquad \mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0},$$

its conditional expectation is

$$\begin{aligned} \mathbb{E}[\widetilde{\mathbf{r}} \mid \mathbf{X}] &= A_\lambda^{-1} \mathbf{X}^\top \mathbf{X} \mathbf{r}^\star \\ &= \left(I_d - \lambda A_\lambda^{-1}\right) \mathbf{r}^\star. \end{aligned} \tag{21}$$

The coordinate-wise reference bias is therefore

$$b_{0,j} := \mathbb{E}[\widetilde{r}_j \mid \mathbf{X}] - r_j^\star = -\lambda e_j^\top A_\lambda^{-1} \mathbf{r}^\star. \tag{22}$$

Under the reward bound $\|\mathbf{r}^\star\|_\infty \le r_{\max}$, Hölder's inequality gives

$$|b_{0,j}| \le \lambda r_{\max} \left\| e_j^\top A_\lambda^{-1} \right\|_1 =: B_j. \tag{23}$$

This bound reflects local identification. When coordinate j is well identified by the aggregate trajectories, the corresponding row of A_λ^{-1} can be small, particularly for a small ridge penalty. If j is completely unvisited, then $\mathbf{X}_{:,j} = \mathbf{0}$ and

$$A_\lambda^{-1} e_j = \lambda^{-1} e_j.$$

Consequently,

$$\widetilde{r}_j = 0, \qquad b_{0,j} = -r_j^\star, \qquad B_j = r_{\max}.$$

Thus, a zero ridge prediction at an unvisited coordinate is accompanied by maximal reference-bias uncertainty rather than being treated as a reliable zero reward.

Variance-corrected disagreement. For surrogate m, define

$$b_{m,j} := \mathbb{E}[\widehat{r}_{m,j} \mid \mathbf{X}] - r_j^\star$$

and let

$$\Delta_{m,j} := \widehat{r}_{m,j} - \widetilde{r}_j.$$

The conditional mean of this disagreement is the relative bias

$$\delta_{m,j} := \mathbb{E}[\Delta_{m,j} \mid \mathbf{X}] = b_{m,j} - b_{0,j}.$$

If $\Sigma_{j,m\ell}$ denotes the conditional sampling covariance between the predictions of models m and ℓ at coordinate j, then

$$v_{\Delta,m,j} := \mathrm{Var}(\Delta_{m,j} \mid \mathbf{X}) = \Sigma_{j,mm} + \Sigma_{j,00} - 2\Sigma_{j,0m}. \tag{24}$$

It follows that

$$\mathbb{E}[\Delta_{m,j}^2 \mid \mathbf{X}] = \delta_{m,j}^2 + v_{\Delta,m,j}. \tag{25}$$

Equation equation 25 motivates the method-of-moments proxy

$$\widehat{d}_{m,j} := \sqrt{\left[\Delta_{m,j}^2 - \widehat{v}_{\Delta,m,j}\right]_+} \tag{26}$$

for the relative-bias magnitude $|\delta_{m,j}|$. The positive part is necessary because an observed squared disagreement may be smaller than its estimated sampling variance.

From relative bias to surrogate bias. The reference model is useful because disagreement identifies relative bias:

$$b_{m,j} = \delta_{m,j} + b_{0,j}.$$

Although $b_{0,j}$ is unknown, Equation equation 23 restricts it to $[-B_j, B_j]$. The reverse triangle inequality and triangle inequality therefore imply

$$[|\delta_{m,j}| - B_j]_+ \le |b_{m,j}| \le |\delta_{m,j}| + B_j. \tag{27}$$

These endpoints are sharp given only $|b_{0,j}| \le B_j$: the lower endpoint attributes as much disagreement as possible to reference bias, whereas the upper endpoint assigns the reference uncertainty against the surrogate.

Replacing $|\delta_{m,j}|$ with $\widehat{d}_{m,j}$ yields the plug-in endpoints

$$\widehat{L}_{m,j} = \left[\widehat{d}_{m,j} - B_j\right]_+, \qquad \widehat{U}_{m,j} = \widehat{d}_{m,j} + B_j. \tag{28}$$

We interpolate between these choices through

$$\widehat{b}_{m,j}(\beta_B) = \left[\widehat{d}_{m,j} + \beta_B B_j\right]_+, \qquad \beta_B \in [-1, 1]. \tag{29}$$

At $\beta_B = -1$, Equation equation 29 equals $\widehat{L}_{m,j}$, giving the surrogate the benefit of the doubt whenever disagreement could be explained by bias in the reference. At $\beta_B = +1$, it equals $\widehat{U}_{m,j}$, giving the reference the benefit of the doubt and penalizing the surrogate for both disagreement and possible reference bias. The choice $\beta_B = 0$ assigns the estimated relative-bias magnitude directly to the surrogate. Intermediate values provide a monotone interpolation governed by the assumed trustworthiness of the surrogate and the application context.

The interval in Equation equation 27 is a population statement conditional on $\mathbf{X}$. Its plug-in counterpart is not a confidence interval: $\widehat{d}_{m,j}$, $\widehat{v}_{\Delta,m,j}$, and therefore both endpoints are estimated with sampling error. The construction also estimates bias magnitude rather than bias sign and does not rely on cancellation between unknown biases. Accordingly, β_B is a modeling or tuning parameter, fixed before evaluation, rather than a confidence level.

C Proofs for Theoretical Analysis

This appendix proves Proposition 2, Lemma 2, and Theorem 1. Throughout, all fitted models use the same labels. The covariance estimate is the centered residual sandwich matrix specified below. The argument uses its algebraic form, without identifying it with the true sampling covariance.

C.1 Proof of Proposition 2

Proof. Conditional on $\mathbf{X}$, the oracle weights are fixed, so by Proposition 1 the conditional MSE of any weight vector $\mathbf{w}$ in the simplex $\mathcal{W} = \{\mathbf{w} \ge \mathbf{0} : \mathbf{1}^\top \mathbf{w} = 1\}$ equals $\mathbf{w}^\top \mathbf{R}_j \mathbf{w}$, and

$$\mathbb{E}\Big[\big((\mathbf{w}_j^{\mathrm{oracle}})^\top \widehat{\mathbf{r}}_j - r_j^\star\big)^2 \mid \mathbf{X}\Big] = \min_{\mathbf{w}\in\mathcal{W}} \mathbf{w}^\top \mathbf{R}_j \mathbf{w}.$$

The minimum exists because the objective is continuous on the compact set $\mathcal{W}$. It therefore suffices to exhibit two feasible weight vectors whose objective values are the two terms in the stated minimum. Using $\mathbf{R}_j = \mathbf{\Sigma}_j + \mathbf{b}_j \mathbf{b}_j^\top$,

$$\mathbf{w}^\top \mathbf{R}_j \mathbf{w} = \mathbf{w}^\top \mathbf{\Sigma}_j \mathbf{w} + (\mathbf{w}^\top \mathbf{b}_j)^2. \tag{30}$$

Reference-only. The vector $\mathbf{w} = e_0$ is feasible and gives $e_0^\top \mathbf{R}_j e_0 = R_{j,00}$.

Equal weights on the surrogates. Let $\bar{\mathbf{w}} = M^{-1} \sum_{m=1}^M e_m$, which is feasible. For the bias term in equation 30,

$$(\bar{\mathbf{w}}^\top \mathbf{b}_j)^2 = \Big(\frac{1}{M}\sum_{m=1}^M b_{m,j}\Big)^2 \le \Big(\frac{1}{M}\sum_{m=1}^M |b_{m,j}|\Big)^2 \le \bar{b}^2.$$

For the variance term, split the double sum into diagonal and off-diagonal parts. The M diagonal entries satisfy $\Sigma_{j,mm} \le \sigma_s^2$, and each of the $M(M-1)$ off-diagonal entries satisfies, because $\rho \ge 0$,

$$\Sigma_{j,m\ell} \le \rho\sqrt{\Sigma_{j,mm}\Sigma_{j,\ell\ell}} \le \rho\sigma_s^2.$$

Hence

$$\bar{\mathbf{w}}^\top \mathbf{\Sigma}_j \bar{\mathbf{w}} = \frac{1}{M^2}\Big[\sum_{m=1}^{M} \Sigma_{j,mm} + \sum_{m\neq\ell} \Sigma_{j,m\ell}\Big] \le \frac{M\sigma_s^2 + M(M-1)\rho\sigma_s^2}{M^2} = \sigma_s^2\Big(\rho + \frac{1-\rho}{M}\Big).$$

Combining the two terms, $\bar{\mathbf{w}}^\top \mathbf{R}_j \bar{\mathbf{w}} \le \bar{b}^2 + \sigma_s^2\big(\rho + (1-\rho)/M\big)$.

Since the oracle value is the minimum over $\mathcal{W}$, it is at most the smaller of the two feasible values, which proves the bound. If $\bar{b}^2 + \sigma_s^2(\rho + (1-\rho)/M) < R_{j,00}$, the minimum is attained strictly below $R_{j,00}$, proving the final claim. The same argument applies to any subset of the surrogates, with M replaced by the size of the subset, because equal weights on a subset are also feasible. □

C.2 Full specification and confidence constants

We analyze tabular collection with additive aggregate feedback, a stationary transition kernel P, initial-state distribution μ_0, and $|r^\star(s,a)| \le r_{\max}$. Write $S = |\mathcal{S}|$, $A = |\mathcal{A}|$, and $d = SA$. In collection block k, the algorithm executes one deterministic mapping $\pi_k : \mathcal{S} \to \mathcal{A}$: $a_{k,h} = \pi_k(s_{k,h})$ for $h = 1, \dots, H_k$. The block length $1 \le H_k \le H$ and the feedback-segment boundaries are selected before collecting that block. Each block is partitioned into at most $L_{\mathrm{seg}} \le H$ feedback segments, so that every step of the block belongs to exactly one labeled segment, and all its labels are available before the next update. The models, mixture weights, and policy are held fixed within the block. The initial state of each block is drawn from μ_0; the theorem measures the finite-horizon return from this distribution.

Let $N = \sum_{k=1}^{K} H_k$ be the realized number of collection steps, $t_k = \sum_{k' \le k} H_{k'}$, and $t_0 = 0$. Fix a known deterministic budget $\bar{N} \ge N$, for example $\bar{N} = KH$. An initial dataset of at most N_0 steps may be collected by history-dependent policies in the same MDP that may use all past trajectories but not the labels of the trajectory currently being generated, with trajectories of length at most H, each partitioned into at most L_{seg} feedback segments, and the same noise assumptions. Put $\bar{T} = N_0 + \bar{N}$. Let $N_k(j)$ count visits to pair j before block k, including initialization, write $N_k = (N_k(j))_{j=1}^{d}$, and let X_k, y_k contain the completed aggregate data. When all blocks have the same arbitrary length H, $N = \bar{N} = HK$. The planner's transition model $\widehat{P}_k(\cdot \mid j)$ is the empirical frequency of the observed next states over these $N_k(j)$ visits; at an unvisited pair it may be any probability distribution. The matrix $A_k = \lambda I_d + X_k^\top X_k$ is A_λ of equation 5 computed from X_k. Throughout, $\mathcal{F}_{k-1}$ denotes the information available before block k starts: the initialization, all completed blocks and their labels, the chosen block length H_k and feedback boundaries, the fitted models, and the selected mapping π_k, but not the initial state of block k. Block k is collected under a fixed $\beta_B \in [-1, 1]$ in the bias proxies equation 10.

The result covers fixed-feature linear surrogates fitted by ridge regression and the trajectory-clustered residual sandwich estimator. Let $\Phi_m \in \mathbb{R}^{d \times p_m}$ contain the surrogate features, so $Z_{m,k} = X_k \Phi_m$, and let $\lambda_m > 0$ be its fixed ridge penalty. The reference has $\Phi_0 = I_d$ and $\lambda_0 = \lambda > 0$. Fixed external reward tables may be included with zero covariance rows and columns. Let $\mathcal{L}$ index the fitted linear models, including the reference. No surrogate is required to represent $r^\star$. The implemented estimators and adaptive weights are specified in Appendix C.4.

Let Π_{stat} be the class of deterministic stationary mappings and define their finite-horizon returns

$$J_h(\pi; r, P) := \mathbb{E}_{\mu_0, P, \pi}\left[\sum_{t=1}^{h} r(s_t, \pi(s_t))\right]. \tag{31}$$

The holding condition in Assumption 2 is

$$P(s \mid s, a) \ge p \quad \text{for all state–action pairs that may be selected.} \tag{32}$$

Assumption 1 permits heteroskedasticity and within-trajectory dependence subject to its joint sub-Gaussian bound. It does not condition on future adaptive designs. The parameter σ is the aggregate-label noise scale, including any segment-length dependence. The holding condition permits sparse

transitions, absorbing states, reducible chains, and unequal visitation; it requires no positive visitation rate on every pair.

Fix $\delta \in (0,1)$. Define b_D and η by equation 37 below. For the same δ, define deterministic confidence constants

$$Q := \max_{m \in \mathcal{L}} \max \left\{ 1, \frac{\lambda \|\Phi_m\|_{\mathrm{op}}^2}{\lambda_m} \right\}, \tag{33}$$

$$\alpha_N := \sigma \sqrt{d \log\left(1 + \frac{H\bar{T}}{\lambda d}\right) + 2 \log \frac{6}{\delta}} + r_{\max} \sqrt{\lambda d}, \tag{34}$$

$$R_N := \alpha_N + \sigma \sqrt{2 L_{\mathrm{seg}} \log \frac{12\bar{T}}{\delta}}, \qquad C_N := R_N \sqrt{\frac{Q \max\{1, \lambda^{-1}\}}{\eta}}. \tag{35}$$

Their dependence on the collection budget is at most logarithmic when the other parameters, including H, N_0, p, are fixed.

$$\ell_N := \log \frac{12 S^2 A(\bar{T}+1)}{\delta}, \qquad c_{P,N} := 4\sqrt{S\ell_N}, \qquad \beta_r = 1, \qquad \beta_P = 4C_N + H r_{\max} c_{P,N}. \tag{36}$$

The return model used for planning is $\widehat{J}_k(\pi) = J_{H_k}(\pi; r_{\mathrm{plan},k}, \widehat{P}_k)$. Since H_k is chosen before collection, t_k is known when planning. For equal block lengths, Assumption 3 permits a gap $c_{\mathrm{opt}} \sqrt{H/k}$. The explicit regret bound appears in equation 75.

C.3 Design and adaptive-mixing lemmas

Lemma 1 (Identifiability derived from holding probabilities)**.** *Fix $\delta \in (0,1)$ and define*

$$b_D = 5H^2 \log \frac{6d}{\delta} + \frac{p}{2} N_0, \qquad \eta = \frac{p\lambda}{2(\lambda + b_D)}. \tag{37}$$

Under Assumption 2 and the collection specification, with probability at least $1 - \delta/6$, simultaneously at every update,

$$A_k := \lambda I_d + X_k^\top X_k \succeq \eta \, \mathrm{diag}_{j=1}^d\big(\lambda + N_k(j)\big). \tag{38}$$

For one aggregate label in a block, its count vector $\mathbf{x}_k$ satisfies

$$\mathbb{E}[\mathbf{x}_k \mathbf{x}_k^\top \mid \mathcal{F}_{k-1}] \succeq p \, \mathbb{E}[\mathrm{diag}(\mathbf{x}_k) \mid \mathcal{F}_{k-1}], \tag{39}$$

with $\mathcal{F}_{k-1}$ as defined in Appendix C.2. This population constant is independent of block length. For several segments, the same comparison holds between the sum of their count outer products and their total visit counts.

Lemma 2 (Statistical control and stability of adaptive mixing)**.** *Under Assumptions 1 and 2 and the fitting and collection specifications, with probability at least $1 - \delta/2$, simultaneously for every block k and pair j, with $q_{k,j} := C_N / \sqrt{N_k(j) + 1}$,*

$$\max \left\{ |\widetilde{r}_{k,j} - r_j^\star|, \; B_{k,j}, \; \max_{0 \le m \le M} \sqrt{\widehat{\Sigma}_{k,j,mm}} \right\} \le q_{k,j}. \tag{40}$$

For the implemented simplex weights, this implies

$$\sum_{m=1}^{M} \widehat{w}_{m,k,j} \sqrt{\widehat{v}_{\Delta,m,k,j}} \le 2 q_{k,j}, \tag{41}$$

$$b_{r,k,j} \le \sqrt{2}\, q_{k,j}, \qquad |\widehat{r}_{\mathrm{mix},k,j} - r_j^\star| \le b_{r,k,j} + 4 q_{k,j}. \tag{42}$$

The predictions, covariance estimates, and weights may use the same data. Neither covariance calibration nor vanishing surrogate bias is required. The uncertainty inequalities follow from the noise and holding assumptions; they are not additional assumptions.

C.4 Notation and the implemented quantities

Let $\mathcal{H}_g$ contain all data through trajectory g, and use the nested information sets $\mathcal{H}_{g-1} \subseteq \mathcal{G}_g \subseteq \mathcal{H}_g$, where $\mathcal{G}_g$ additionally reveals the current trajectory and its segmentation but not its label vector. Label vectors can be exposed at trajectory end in this analysis because the policy is held fixed during the block. The sequential noise assumption is imposed on this exposure order. Let X_k, y_k stack all completed feedback segments before collection block k, including initialization. Write $n_{k,j} = N_k(j)$ and $u_{k,j} = (n_{k,j}+1)^{-1/2}$. Thus $N_k(j)$ is the count $N(s,a)$ of equation 12 evaluated before block k, and $\beta_P u_{k,j} = \beta_P b_P(s,a)$.

For $m \in \mathcal{L}$, let $\phi_m(j) = \Phi_m^\top e_j$ and define

$$
\begin{aligned}
Z_{m,k} &= X_k \Phi_m, \qquad A_{m,k} = Z_{m,k}^\top Z_{m,k} + \lambda_m I, \\
\widehat{r}_{m,k,j} &= \phi_m(j)^\top A_{m,k}^{-1} Z_{m,k}^\top y_k = a_{m,k,j}^\top y_k, \qquad a_{m,k,j} = Z_{m,k} A_{m,k}^{-1} \phi_m(j).
\end{aligned} \tag{43}
$$

For $m = 0$, $A_{0,k} = A_k$ and $\widehat{r}_{0,k,j} = \widetilde{r}_{k,j}$. Let $a_{m,k,j,g}$ denote the sub-vector of $a_{m,k,j}$ on the rows of trajectory g; since the trajectories partition the rows of $Z_{m,k}$, $\sum_g \|a_{m,k,j,g}\|_2^2 = \|a_{m,k,j}\|_2^2$. Let G_k be the number of completed trajectories at this update and set

$$
\begin{aligned}
\widehat{\varepsilon}_{k,g} &:= y_g - X_g \widetilde{r}_k, \qquad \widehat{q}_{m,k,j,g} := a_{m,k,j,g}^\top \widehat{\varepsilon}_{k,g}, \\
\bar{\widehat{q}}_{m,k,j} &:= G_k^{-1} \sum_{g=1}^{G_k} \widehat{q}_{m,k,j,g}, \\
\widehat{\Sigma}_{k,j,m\ell} &:= \sum_{g=1}^{G_k} (\widehat{q}_{m,k,j,g} - \bar{\widehat{q}}_{m,k,j})(\widehat{q}_{\ell,k,j,g} - \bar{\widehat{q}}_{\ell,k,j}).
\end{aligned} \tag{44}
$$

For a fixed external surrogate, set its contributions $\widehat{q}_{m,k,j,g}$ to zero. If $G_k = 0$, set $\widehat{\Sigma}_{k,j} = 0$. Equation equation 44 is a sum of outer products, and hence $\widehat{\Sigma}_{k,j} \succeq 0$ for every realized dataset.

For completeness, the bias proxies and optimization used in the proof are, for the fixed $\beta_B \in [-1,1]$ of equation 10,

$$
\begin{aligned}
B_{k,j} &= \lambda r_{\max} \|e_j^\top A_k^{-1}\|_1, \\
\Delta_{m,k,j} &= \widehat{r}_{m,k,j} - \widetilde{r}_{k,j}, \qquad \widehat{v}_{\Delta,m,k,j} = \widehat{\Sigma}_{k,j,mm} + \widehat{\Sigma}_{k,j,00} - 2\widehat{\Sigma}_{k,j,0m}, \\
\widehat{b}_{0,k,j} &= B_{k,j}, \qquad \widehat{b}_{m,k,j} = \left[\sqrt{[\Delta_{m,k,j}^2 - \widehat{v}_{\Delta,m,k,j}]_+} + \beta_B B_{k,j}\right]_+ \quad (m \geq 1), \\
\widehat{R}_{k,j} &= \widehat{\Sigma}_{k,j} + \widehat{b}_{k,j}\widehat{b}_{k,j}^\top, \qquad \widehat{w}_{k,j} \in \arg\min_{w \in \mathcal{W}} w^\top \widehat{R}_{k,j} w, \quad \mathcal{W} = \{w \geq 0 : \mathbf{1}^\top w = 1\}, \\
\widehat{r}_{\mathrm{mix},k,j} &= \sum_{m=0}^{M} \widehat{w}_{m,k,j} \widehat{r}_{m,k,j}, \qquad b_{r,k,j} = \sqrt{\widehat{w}_{k,j}^\top \widehat{R}_{k,j} \widehat{w}_{k,j}}.
\end{aligned} \tag{45}
$$

The minimum exists because the objective is continuous on the compact simplex. Any measurable choice of an exact minimizer is admissible; uniqueness is unnecessary. The planning reward is

$$
r_{\mathrm{plan},k,j} = \widehat{r}_{\mathrm{mix},k,j} + \beta_r b_{r,k,j} + \beta_P u_{k,j}. \tag{46}
$$

C.5 Identifiability from holding probabilities

We prove Lemma 1 without assuming a spectral lower bound on the aggregate design. First let $s_0, \ldots, s_{h-1}$ be a time-homogeneous chain with transition matrix $\mathbf{Q}$ satisfying $\mathbf{Q}(s,s) \geq p$. For any function f and any initial distribution, we claim

$$
\mathbb{E}\left[\left(\sum_{t=0}^{h-1} f(s_t)\right)^2\right] \geq p\, \mathbb{E} \sum_{t=0}^{h-1} f(s_t)^2. \tag{47}
$$

For $h = 1$ this is immediate. For $h \geq 2$, write $\mathbf{Q} = pI + (1-p)\widetilde{\mathbf{Q}}$, where $\widetilde{\mathbf{Q}} = (\mathbf{Q} - pI)/(1-p)$ is stochastic. Realize a transition by an independent Bernoulli variable χ_t with mean p: stay if $\chi_t = 1$,

otherwise sample from $\widetilde{\mathbf{Q}}$. This auxiliary representation leaves the original transition law unchanged and introduces no algorithmic randomization.

Set

$$\mathsf{F} = \sum_{t=0}^{h-1} f(s_t), \quad \mathsf{Z} = \mathbb{E}\sum_{t=0}^{h-1} f(s_t)^2, \quad \mathsf{z} = \mathbb{E} f(s_{h-1})^2, \quad \mathsf{M} = \sum_{t=0}^{h-2} f(s_t)(\chi_t - p).$$

The summands of M are martingale differences, so

$$\mathbb{E}\mathsf{M}^2 = p(1-p)(\mathsf{Z} - \mathsf{z}). \tag{48}$$

Let $\Psi_m = \sum_{i=0}^{m-1} \mathbf{Q}^i f$ and let $\mathcal{I}_t$ reveal the chain and auxiliary transition randomness through s_t, before χ_t. The past contribution to F has zero covariance with $\chi_t - p$, whereas

$$\begin{aligned}\mathbb{E}[(\chi_t - p)\mathsf{F} \mid \mathcal{I}_t] &= p(1-p)(I - \widetilde{\mathbf{Q}})\Psi_{h-1-t}(s_t) \\ &= p(I - \mathbf{Q})\Psi_{h-1-t}(s_t) = p\big(f(s_t) - \mathbf{Q}^{h-1-t} f(s_t)\big).\end{aligned}$$

Multiplying by $f(s_t)$, taking expectations, and summing yields

$$\mathbb{E}[\mathsf{F}\mathsf{M}] = p\sum_{t=0}^{h-2} \mathbb{E}[f(s_t)^2 - f(s_t) f(s_{h-1})] = p\big(\mathsf{Z} - \mathbb{E}[\mathsf{F} f(s_{h-1})]\big).$$

With $\mathsf{W} = \mathsf{M} + p f(s_{h-1})$, this says $\mathbb{E}[\mathsf{F}\mathsf{W}] = p\mathsf{Z}$. The triangle inequality in L^2, followed by Cauchy–Schwarz, gives

$$\|\mathsf{W}\|_{L^2} \le \sqrt{p(1-p)(\mathsf{Z} - \mathsf{z})} + p\sqrt{\mathsf{z}} \le \sqrt{\big(p(1-p) + p^2\big)\mathsf{Z}} = \sqrt{p\mathsf{Z}}.$$

If $\mathsf{Z} > 0$, then $p^2\mathsf{Z}^2 = (\mathbb{E}[\mathsf{F}\mathsf{W}])^2 \le \mathbb{E}[\mathsf{F}^2]\mathbb{E}[\mathsf{W}^2] \le p\mathsf{Z}\mathbb{E}[\mathsf{F}^2]$, proving equation 47. The case $\mathsf{Z} = 0$ is trivial.

Conditional on the past and the chosen mapping in a collection block, the state chain has $\mathbf{Q}(s, s') = P(s' \mid s, \pi_k(s))$. For $v \in \mathbb{R}^d$, apply equation 47 to $f(s) = v(s, \pi_k(s))$. This gives equation 39. With several preselected feedback segments, apply the same argument conditionally at the beginning of each segment and use iterated expectation. Writing their count vectors as $\mathbf{x}_{k,i}$, define

$$\Gamma_k = \sum_i \mathbf{x}_{k,i}\mathbf{x}_{k,i}^\top, \qquad D_k' = \operatorname{diag}\left(\sum_i \mathbf{x}_{k,i}\right).$$

We obtain $\mathbb{E}[\Gamma_k \mid \mathcal{F}_{k-1}] \succeq p\,\mathbb{E}[D_k' \mid \mathcal{F}_{k-1}]$.

To pass to the realized design, put $\Upsilon_k = (p/2)D_k' - \Gamma_k$. Then, conditionally on $\mathcal{F}_{k-1}$,

$$\mathbb{E}\Upsilon_k \preceq -\tfrac{1}{2}\mathbb{E}\Gamma_k.$$

Because the segment lengths sum to at most H, $\|\Gamma_k\|_{\mathrm{op}} \le H^2$, $(\Gamma_k)^2 \preceq H^2\Gamma_k$, and $(D_k')^2 \preceq H D_k'$. Using $(U - V)^2 \preceq 2U^2 + 2V^2$ for self-adjoint U, V gives

$$\begin{aligned}\mathbb{E}\Upsilon_k^2 &\preceq \frac{p^2 H}{2}\mathbb{E}D_k' + 2H^2\mathbb{E}\Gamma_k \\ &\preceq \left(\frac{pH}{2} + 2H^2\right)\mathbb{E}\Gamma_k \preceq \frac{5}{2}H^2\mathbb{E}\Gamma_k.\end{aligned}$$

Also $\|\Upsilon_k\|_{\mathrm{op}} \le 3H^2/2$. For $\vartheta = 1/(5H^2)$, the scalar bound $\mathrm{e}^u \le 1 + u + u^2$ on $[-3/10, 3/10]$, applied spectrally, implies

$$\mathbb{E}[\mathrm{e}^{\vartheta\Upsilon_k} \mid \mathcal{F}_{k-1}] \preceq I_d + \vartheta\mathbb{E}\Upsilon_k + \vartheta^2\mathbb{E}\Upsilon_k^2 \preceq I_d.$$

Golden–Thompson now makes $\Xi_n = \operatorname{tr}\exp(\vartheta\sum_{k=1}^n \Upsilon_k)$ a nonnegative supermartingale with $\Xi_0 = d$: its conditional expectation is at most

$$\operatorname{tr}\left[\mathrm{e}^{\vartheta\sum_{k<n}\Upsilon_k}\mathbb{E}[\mathrm{e}^{\vartheta\Upsilon_n} \mid \mathcal{F}_{n-1}]\right] \le \Xi_{n-1}.$$

Ville's inequality consequently gives, with probability at least $1 - \delta/6$, simultaneously for every n,

$$\sum_{k=1}^{n} \Gamma_k \succeq \frac{p}{2} \sum_{k=1}^{n} D'_k - 5H^2 \log(6d/\delta) I_d. \tag{49}$$

This is a direct matrix-exponential concentration argument; see Tropp (2011) for the general framework.

The initialization contributes a positive semidefinite Gram matrix and a count diagonal bounded above by $N_0 I_d$. Thus adding it to equation 49 gives $X_k^\top X_k \succeq (p/2)\,\mathrm{diag}(N_k) - b_D I_d$ at all updates. We also have $A_k \succeq \lambda I_d$. Taking the convex combination with weight $\omega = \lambda/(\lambda + b_D)$ on $A_k \succeq (\lambda - b_D) I_d + (p/2)\,\mathrm{diag}(N_k)$ yields

$$A_k \succeq \frac{\lambda^2}{\lambda + b_D} I_d + \frac{p\lambda}{2(\lambda + b_D)}\,\mathrm{diag}(N_k) \succeq \eta\big(\lambda I_d + \mathrm{diag}(N_k)\big).$$

This proves Lemma 1. Denote its event by $\mathcal{E}_D$.

C.6 Self-normalized concentration for trajectory labels

We give the cluster-vector version of the exponential-mixture argument underlying self-normalized linear-regression bounds (Abbasi-Yadkori et al., 2011). This derivation makes the conditioning under adaptive collection explicit.

Lemma 3 (Cluster self-normalized inequality)**.** *Let* $\Lambda_g = \lambda I_d + \sum_{g' \le g} X_{g'}^\top X_{g'}$ *and* $S_g = \sum_{g' \le g} X_{g'}^\top \varepsilon_{g'}$*. Under Assumption 1, for every* $\delta' \in (0,1)$*, with probability at least* $1 - \delta'$*, simultaneously over all completed trajectory prefixes,*

$$\|S_g\|^2_{\Lambda_g^{-1}} \le \sigma^2 \left[\log \frac{\det \Lambda_g}{\det(\lambda I_d)} + 2 \log \frac{1}{\delta'}\right]. \tag{50}$$

Proof. First suppose $\sigma > 0$. For a fixed $z \in \mathbb{R}^d$, define

$$\mathsf{N}_g(z) = \exp\left\{ \frac{z^\top S_g}{\sigma} - \frac{1}{2} z^\top (\Lambda_g - \lambda I_d) z \right\}.$$

Conditioning first on $\mathcal{G}_g$ and applying equation 13 with $v = X_g z/\sigma$ gives $\mathbb{E}[\mathsf{N}_g(z) \mid \mathcal{G}_g] \le \mathsf{N}_{g-1}(z)$. The tower property therefore makes $\mathsf{N}_g(z)$ a nonnegative supermartingale with respect to $\mathcal{H}_g$. Integrate over an independent $z \sim N(0, \lambda^{-1} I_d)$. Tonelli's theorem preserves the supermartingale property, and Gaussian integration gives

$$\overline{\mathsf{N}}_g = \left(\frac{\det(\lambda I_d)}{\det \Lambda_g}\right)^{1/2} \exp\left\{ \frac{\|S_g\|^2_{\Lambda_g^{-1}}}{2\sigma^2} \right\}, \qquad \overline{\mathsf{N}}_0 = 1.$$

Ville's inequality yields $\Pr(\sup_g \overline{\mathsf{N}}_g \ge 1/\delta') \le \delta'$. Taking logarithms proves equation 50. If $\sigma = 0$, the conditional moment-generating-function bound implies $\varepsilon_g = 0$ almost surely, so the result is immediate. □

For every completed segment i, its nonnegative count vector $\mathbf{x}_i$ satisfies $\|\mathbf{x}_i\|_1 = T_i \le H$. Because segments do not overlap and at most $\bar{T}$ primitive steps enter the data,

$$\mathrm{tr}(X_k^\top X_k) = \sum_i \|\mathbf{x}_i\|_2^2 \le \sum_i T_i^2 \le H \sum_i T_i \le H\bar{T}.$$

The arithmetic–geometric mean inequality for the eigenvalues then gives

$$\log \frac{\det A_k}{\det(\lambda I_d)} \le d \log\left(1 + \frac{H\bar{T}}{\lambda d}\right). \tag{51}$$

Thus no additional assumption limiting within-segment occupancy is needed for this determinant bound.

The exact ridge identity is

$$\widetilde{\mathbf{r}}_k - \mathbf{r}^\star = A_k^{-1} X_k^\top \varepsilon_k - \lambda A_k^{-1} \mathbf{r}^\star, \tag{52}$$

where ε_k stacks the noise vectors already observed. Apply Lemma 3 with $\delta' = \delta/6$ and use $A_k \succeq \lambda I_d$ and $\|\mathbf{r}^\star\|_2 \le r_{\max}\sqrt{d}$. With probability at least $1 - \delta/6$, for every k,

$$\|\widetilde{\mathbf{r}}_k - \mathbf{r}^\star\|_{A_k} \le \|X_k^\top \varepsilon_k\|_{A_k^{-1}} + \lambda \|\mathbf{r}^\star\|_{A_k^{-1}} \le \alpha_N. \tag{53}$$

Call this event $\mathcal{E}_{\mathrm{ridge}}$. On this event, weighted Cauchy–Schwarz yields

$$|\widetilde{r}_{k,j} - r_j^\star| \le \alpha_N \sqrt{(A_k^{-1})_{jj}}. \tag{54}$$

The ridge-bias proxy is controlled pathwise, since $A_k^{-2} \preceq \lambda^{-1} A_k^{-1}$:

$$B_{k,j} \le \lambda r_{\max}\sqrt{d}\, \|A_k^{-1} e_j\|_2 \le r_{\max}\sqrt{\lambda d}\sqrt{(A_k^{-1})_{jj}} \le \alpha_N \sqrt{(A_k^{-1})_{jj}}. \tag{55}$$

This bound concerns the regularization term in equation 52; it does not require interpreting that term as the conditional bias given the full adaptive design.

C.7 A LEVERAGE COMPARISON FOR ALL LINEAR SURROGATES

Lemma 4 (Ridge leverage transfer)**.** *For every realized dataset, every fitted linear model m, and every coordinate j, let $Q_m = \max\{1, \lambda\|\Phi_m\|_{\mathrm{op}}^2/\lambda_m\}$. Then*

$$\|a_{m,k,j}\|_2^2 \le \phi_m(j)^\top A_{m,k}^{-1} \phi_m(j) \le Q_m (A_k^{-1})_{jj}. \tag{56}$$

This conclusion does not require realizability or full column rank of the surrogate feature matrix.

Proof. Suppress k, so $A = X^\top X + \lambda I_d$. Since $Z_m = X\Phi_m$,

$$\Phi_m^\top A \Phi_m = \Phi_m^\top X^\top X \Phi_m + \lambda \Phi_m^\top \Phi_m \preceq Q_m(\Phi_m^\top X^\top X \Phi_m + \lambda_m I) = Q_m A_m.$$

Consequently $U = A^{1/2}\Phi_m A_m^{-1/2}$ satisfies $U^\top U \preceq Q_m I$, hence $UU^\top \preceq Q_m I$ and

$$\Phi_m A_m^{-1} \Phi_m^\top \preceq Q_m A^{-1}.$$

Also $Z_m^\top Z_m \preceq A_m$, so

$$\|a_{m,j}\|_2^2 = \phi_m(j)^\top A_m^{-1} Z_m^\top Z_m A_m^{-1} \phi_m(j) \le \phi_m(j)^\top A_m^{-1} \phi_m(j) \le Q_m e_j^\top A^{-1} e_j.$$

All inverses exist by positive ridge regularization. □

C.8 CONTROLLING THE COMPUTED SANDWICH COVARIANCE

Each coordinate of ε_g is conditionally σ-sub-Gaussian by equation 13. There are at most $\bar{T}$ labels in total. A union bound and the tower property give an event $\mathcal{E}_{\mathrm{noise}}$ of probability at least $1 - \delta/6$ on which

$$\max_g \|\varepsilon_g\|_2 \le \sigma\sqrt{2L_{\mathrm{seg}} \log(12\bar{T}/\delta)}. \tag{57}$$

No independence of the coordinates is used. On $\mathcal{E}_{\mathrm{ridge}} \cap \mathcal{E}_{\mathrm{noise}}$, take any past trajectory g in the dataset at update k. Since $X_g^\top X_g \preceq A_k$, its reference residual obeys

$$\begin{aligned} \|\widehat{\varepsilon}_{k,g}\|_2 &\le \|\varepsilon_g\|_2 + \|X_g(\mathbf{r}^\star - \widetilde{\mathbf{r}}_k)\|_2 \\ &\le \|\varepsilon_g\|_2 + \|\mathbf{r}^\star - \widetilde{\mathbf{r}}_k\|_{A_k} \le R_N. \end{aligned} \tag{58}$$

This holds for every refit, including when the same trajectory is reused in all later datasets.

Centering a vector reduces its squared Euclidean norm. Therefore, using equation 44, Cauchy–Schwarz within each cluster, and Lemma 4,

$$\begin{aligned} \widehat{\Sigma}_{k,j,mm} &\le \sum_g \widehat{q}_{m,k,j,g}^2 \le \sum_g \|a_{m,k,j,g}\|_2^2 \|\widehat{\varepsilon}_{k,g}\|_2^2 \\ &\le R_N^2 \|a_{m,k,j}\|_2^2 \le Q_m R_N^2 (A_k^{-1})_{jj}. \end{aligned} \tag{59}$$

The same bound holds trivially for the zero covariance rows of fixed external surrogates. Thus the estimated variances are controlled without a concentration or consistency claim for the covariance estimator itself, and without a union bound over surrogate models.

On $\mathcal{E}_D$, Lemma 1 and inversion of positive definite matrices imply

$$(A_k^{-1})_{jj} \le \frac{1}{\eta(\lambda + n_{k,j})}. \tag{60}$$

Because $R_N \ge \alpha_N$, $Q \ge 1$, and

$$\frac{1}{\lambda + n} \le \frac{\max\{1, \lambda^{-1}\}}{n+1} \qquad (n \ge 0),$$

equations equation 54, equation 55, and equation 59 prove equation 40 with the deterministic C_N in equation 35. The event $\mathcal{E}_R := \mathcal{E}_D \cap \mathcal{E}_{\mathrm{ridge}} \cap \mathcal{E}_{\mathrm{noise}}$ has probability at least $1 - \delta/2$.

C.9 Pathwise control of the adaptive weights

Lemma 5 (Deterministic mixture inequality)**.** *Fix one update and coordinate, suppress their indices, and suppose* $\widehat{\Sigma} \succeq 0$ *and* $\max\{|\widetilde{r} - r^\star|, B, \max_m \sqrt{\widehat{\Sigma}_{mm}}\} \le q$*. Then the rule equation 45 satisfies equation 41–equation 42, with* $q_{k,j}$ *replaced by* q*.*

Proof. The reference-only weight e_0 is feasible, so optimality gives

$$b_r^2 \le e_0^\top \widehat{R} e_0 = \widehat{\Sigma}_{00} + B^2 \le 2q^2. \tag{61}$$

Positive semidefiniteness and nonnegative bias proxies also give

$$0 \le \widehat{w}^\top \widehat{b} \le \sqrt{\widehat{w}^\top \widehat{R} \widehat{w}} = b_r. \tag{62}$$

For every surrogate, positive semidefiniteness implies

$$0 \le \widehat{v}_{\Delta,m} \le \big(\sqrt{\widehat{\Sigma}_{mm}} + \sqrt{\widehat{\Sigma}_{00}}\big)^2 \le 4q^2.$$

Since $\sum_{m \ge 1} \widehat{w}_m \le 1$, this proves the weighted uncertainty bound. Put $\bar{\Delta}_m = \sqrt{[\Delta_m^2 - \widehat{v}_{\Delta,m}]_+}$. The elementary inequalities

$$|\Delta_m| \le \bar{\Delta}_m + \sqrt{\widehat{v}_{\Delta,m}}, \qquad \bar{\Delta}_m \le [\bar{\Delta}_m + \beta_B B]_+ + B = \widehat{b}_m + B$$

(the second because $\beta_B \ge -1$) yield

$$|\Delta_m| \le \widehat{b}_m + B + \sqrt{\widehat{v}_{\Delta,m}}.$$

Consequently,

$$\begin{aligned} |\widehat{r}_{\mathrm{mix}} - \widetilde{r}| &\le \sum_{m=1}^{M} \widehat{w}_m |\Delta_m| \\ &\le \sum_{m=1}^{M} \widehat{w}_m \widehat{b}_m + B \sum_{m=1}^{M} \widehat{w}_m + \sum_{m=1}^{M} \widehat{w}_m \sqrt{\widehat{v}_{\Delta,m}} \\ &\le b_r + q + 2q = b_r + 3q. \end{aligned}$$

Adding the reference error proves $|\widehat{r}_{\mathrm{mix}} - r^\star| \le b_r + 4q$. Every step is deterministic conditional on the realized fit. □

Applying Lemma 5 on $\mathcal{E}_R$ completes the proof of Lemma 2. In particular, no argument treats the optimized proxy b_r alone as a confidence interval.

C.10 Uniform concentration of empirical transitions

Lemma 6 (Two transition bounds). *There is an event $\mathcal{E}_P$ of probability at least $1-\delta/6$ on which, simultaneously for every update k, pair j, and next state s',*

$$\|\widehat{P}_k(\cdot \mid j) - P(\cdot \mid j)\|_1 \le c_{P,N} u_{k,j}, \tag{63}$$

$$\widehat{P}_k(s' \mid j) \le (1+1/H)P(s' \mid j) + \frac{3H\ell_N}{n_{k,j}+1}. \tag{64}$$

Proof. For each pair, its successive next-state observations have the stationary transition law at that pair. One may expose them as an independent transition stack; adaptive policies only determine how many entries of each stack are used. Apply the scalar Bernstein inequality to every coordinate and every fixed prefix $1 \le n \le \bar{T}$: fix a pair j and a next state s', write $\mathsf{p} = P(s' \mid j)$, and let $\widehat{\mathsf{p}}_n$ be the frequency of s' among the first n entries of the stack. A union bound over at most $S^2A\bar{T}$ choices gives, except on an event of probability at most $2S^2A\bar{T}\,\mathrm{e}^{-\ell_N} \le \delta/6$,

$$|\widehat{\mathsf{p}}_n - \mathsf{p}| \le \sqrt{2\mathsf{p}\ell_N/n} + \frac{2\ell_N}{3n} \le 2\sqrt{\frac{\mathsf{p}\ell_N}{n+1}} + \frac{4\ell_N}{3(n+1)}.$$

Thus the bounds hold also at the random visit counts $n_{k,j}$. Summing over next states and using $\sum_{s'}\sqrt{P(s' \mid j)} \le \sqrt{S}$ yields

$$\|\widehat{P} - P\|_1 \le 2\sqrt{\frac{S\ell_N}{n+1}} + \frac{4S\ell_N}{3(n+1)}.$$

If $S\ell_N/(n+1) \le 1$, this is at most $4\sqrt{S\ell_N/(n+1)}$; otherwise the deterministic bound $\|\widehat{P}-P\|_1 \le 2$ proves equation 63. For the coordinatewise upper bound, use

$$2\sqrt{\frac{\mathsf{p}\ell_N}{n+1}} \le \frac{\mathsf{p}}{H} + \frac{H\ell_N}{n+1}, \qquad H + 4/3 \le 3H.$$

At $n = 0$, both results hold for any probability distribution $\widehat{P}$, since $c_{P,N} \ge 2$ and $3H\ell_N \ge 1$. □

C.11 Optimism and the regret recursion

Work on $\mathcal{E}_R \cap \mathcal{E}_P$. For brevity write $C = C_N$, $c_P = c_{P,N}$, and define

$$a_N = (8+2\sqrt{2})C + Hr_{\max}c_P, \qquad D_N = (8+2\sqrt{2})C + 2Hr_{\max}c_P. \tag{65}$$

The mixture lemma and the prescribed coefficients give the pointwise planning-reward bounds

$$Hr_{\max}c_P u_{k,j} \le r_{\mathrm{plan},k,j} - r^\star_j \le a_N u_{k,j}. \tag{66}$$

Indeed, the lower bound follows from $\widehat{r}_{\mathrm{mix}} - r^\star \ge -b_r - 4Cu$. For the upper bound, $r_{\mathrm{plan}} - r^\star \le 2b_r + (8C + Hr_{\max}c_P)u$ and $b_r \le \sqrt{2}Cu$.

Use stages $h = 1, \dots, H_k$, with terminal values zero at $H_k + 1$. For every stationary mapping π, let $\widehat{V}^\pi_{k,h}$ and $V^\pi_{k,h}$ be its empirical-planning and true finite-horizon values. Although the mapping is stationary, these value functions depend on the number of stages remaining. Since $\|V^\pi_{k,h+1}\|_\infty \le Hr_{\max}$, equation 63 and equation 66 give

$$r_{\mathrm{plan},k,j} - r^\star_j + (\widehat{P}_k - P)V^\pi_{k,h+1}(j) \ge 0.$$

Backward induction in the Bellman evaluation equations therefore yields

$$\widehat{V}^\pi_{k,h}(s) \ge V^\pi_{k,h}(s) \quad \text{for every } k, h, s, \pi \in \Pi_{\mathrm{stat}}. \tag{67}$$

The transition bound is uniform over all bounded vectors, so no union bound over policies is needed. In particular, this statement holds for the data-dependent collection policy.

Let $\pi^\star_k$ maximize the true H_k-step return over Π_{stat}. By equation 67 and Assumption 3,

$$J_{H_k}(\pi^\star_k; r^\star, P) \le \widehat{J}_k(\pi^\star_k) \le \max_{\pi \in \Pi_{\mathrm{stat}}} \widehat{J}_k(\pi)$$

$$\le \widehat{J}_k(\pi_k) + c_{\mathrm{opt}}\frac{H_k}{\sqrt{t_k}}.$$

This is the only step that invokes the planning guarantee. It does not assume a Bellman optimality equation for the restricted policy class.

Set $\Delta^V_{k,h}(s) = \widehat{V}^{\pi_k}_{k,h}(s) - V^{\pi_k}_{k,h}(s)$. It is nonnegative by equation 67. Because $r_{\mathrm{plan},k,j} \le r_{\max} + a_N$ and $V^{\pi_k}_{k,h} \ge -Hr_{\max}$,

$$0 \le \Delta^V_{k,h}(s) \le H(2r_{\max} + a_N) \le HD_N. \tag{68}$$

The last inequality follows from $Hr_{\max}c_P \ge 2r_{\max}$. For $a = \pi_k(s)$ and $j = (s, a)$, policy evaluation gives

$$\begin{aligned}\Delta^V_{k,h}(s) &= r_{\mathrm{plan},k,j} - r^\star_j + (\widehat{P}_k - P)V^{\pi_k}_{k,h+1}(j) + \widehat{P}_k\Delta^V_{k,h+1}(j)\\ &\le D_N u_{k,j} + (1 + 1/H)P\Delta^V_{k,h+1}(j) + \frac{3H^2SD_N\ell_N}{n_{k,j}+1}.\end{aligned}$$

Here equation 63 bounds the term involving the true policy value. The remaining terms use equation 64, nonnegativity of Δ, and equation 68.

With $\mathcal{F}_{k-1}$ as defined in Appendix C.2, unrolling the recursion under P, averaging over μ_0, and using $(1 + 1/H)^{H_k} \le \mathrm{e}$ gives

$$\begin{aligned}&J_{H_k}(\pi^\star_k; r^\star, P) - J_{H_k}(\pi_k; r^\star, P)\\ &\quad\le c_{\mathrm{opt}}\frac{H_k}{\sqrt{t_k}} + \mathrm{e}D_N\,\mathbb{E}\left[\sum_{h=1}^{H_k}\left\{u_{k,j_{k,h}} + \frac{3H^2S\ell_N}{n_{k,j_{k,h}}+1}\right\}\middle|\mathcal{F}_{k-1}\right],\end{aligned} \tag{69}$$

where $j_{k,h} = (s_{k,h}, a_{k,h})$. This expectation is conditional on the available history, not on a future concentration event.

C.12 Predictable costs and block-frozen counts

Lemma 7 (Predictable nonnegative costs)**.** *If $0 \le \varsigma_k \le H$ is measurable after block k and $\bar{\varsigma}_k = \mathbb{E}[\varsigma_k \mid \mathcal{F}_{k-1}]$, then for $\delta' \in (0, 1)$, with probability at least $1 - \delta'$,*

$$\sum_{k=1}^{K}\bar{\varsigma}_k \le 2\sum_{k=1}^{K}\varsigma_k + 2H\log(1/\delta'). \tag{70}$$

Proof. For $z \in [0, 1]$, convexity gives $\mathrm{e}^{-z} \le 1 - cz$, where $c = 1 - \mathrm{e}^{-1} > 1/2$. Therefore

$$\exp\left\{\frac{c}{H}\sum_{k'=1}^{k}\bar{\varsigma}_{k'} - \frac{1}{H}\sum_{k'=1}^{k}\varsigma_{k'}\right\}$$

is a nonnegative supermartingale starting at one: its conditional multiplicative expectation is at most $\mathrm{e}^{c\bar{\varsigma}_k/H}(1 - c\bar{\varsigma}_k/H) \le 1$. Markov's inequality at K, followed by $1/c < 2$, proves the claim. □

Apply this lemma, each time with $\delta' = \delta/6$, to the costs

$$\varsigma_k = \sum_{h=1}^{H_k}\frac{1}{\sqrt{n_{k,j_{k,h}}+1}}, \qquad \varpi_k = \sum_{h=1}^{H_k}\frac{1}{n_{k,j_{k,h}}+1}.$$

Both lie in $[0, H]$. Except on an event of probability at most $\delta/3$, writing $L_\delta = \log(6/\delta)$, we have

$$\begin{aligned}\sum_k \mathbb{E}[\varsigma_k \mid \mathcal{F}_{k-1}] &\le 2\sum_k \varsigma_k + 2HL_\delta,\\ \sum_k \mathbb{E}[\varpi_k \mid \mathcal{F}_{k-1}] &\le 2\sum_k \varpi_k + 2HL_\delta.\end{aligned} \tag{71}$$

The realized costs admit deterministic bounds that respect the block-frozen update schedule. For a fixed pair j, blocks that start with $n_{k,j} < H$ contain fewer than $2H$ visits to that pair in total: before the last such block the count is below H, and that block adds at most H visits. Charge each of these

early visits cost one. At every remaining visit let ν_j denote the number of visits before the current primitive step, including initialization. Then

$$\nu_j + 1 \le n_{k,j} + H \le 2(n_{k,j} + 1), \qquad n_{k,j} \ge H.$$

If N_j counts visits to j in the K collection blocks, $\sum_j N_j = N$. Summing over the remaining visits and using $\sum_{i=1}^{n} i^{-1/2} \le 2\sqrt{n}$ yields

$$\sum_k \varsigma_k \le U_{1/2} := 2Hd + 2\sqrt{2dN}. \tag{72}$$

For the reciprocal cost, the denominators $\nu_j + 1$ of successive late visits are distinct increasing integers at least two. If there are ν' such visits, their reciprocal sum is at most $\sum_{i=1}^{\nu'}(i+1)^{-1} \le \log(\nu'+1) \le \log(N_j+1)$. It follows that

$$\sum_k \varpi_k \le U_1 := 2Hd + 2d\log(N+1). \tag{73}$$

Initial counts can only decrease these costs. No doubling schedule or update within a block is used.

C.13 Completion of the regret proof

Intersect $\mathcal{E}_R$, $\mathcal{E}_P$, and the two events in equation 71. Their total failure probability is at most $\delta/2 + \delta/6 + \delta/3 = \delta$. The planning tolerance has a pathwise cumulative bound. Since $H_k = t_k - t_{k-1}$ and $t_K = N$,

$$\frac{H_k}{\sqrt{t_k}} = \left(\sqrt{t_k} - \sqrt{t_{k-1}}\right)\left(1 + \sqrt{\frac{t_{k-1}}{t_k}}\right) \le 2\left(\sqrt{t_k} - \sqrt{t_{k-1}}\right).$$

Consequently,

$$\sum_{k=1}^{K} c_{\mathrm{opt}} \frac{H_k}{\sqrt{t_k}} \le 2c_{\mathrm{opt}} \sum_{k=1}^{K}\left(\sqrt{t_k} - \sqrt{t_{k-1}}\right) = 2c_{\mathrm{opt}}\sqrt{N}. \tag{74}$$

Summing equation 69 and applying equation 71–equation 73 and equation 74 gives the explicit finite-sample statement

$$\mathrm{Reg}_{\mathrm{stat}}(K) \le \mathrm{e}D_N\left[2U_{1/2} + 2HL_\delta + 3H^2S\ell_N\left(2U_1 + 2HL_\delta\right)\right] + 2c_{\mathrm{opt}}\sqrt{N}. \tag{75}$$

Since $d = SA$, $N = \sum_{k=1}^{K} H_k$, and $D_N = \widetilde{O}(C_N + Hr_{\max}\sqrt{S})$, the right-hand side is

$$\widetilde{O}\Big[(C_N + Hr_{\max}\sqrt{S})\left(\sqrt{SAN} + H^3S^2A\right)\Big] + 2c_{\mathrm{opt}}\sqrt{N},$$

which proves Theorem 1. The bonus coefficient $4C_N$ compensates for the reference and disagreement terms in the mixture lemma. Thus, despite its notation, $\beta_P u_{k,j}$ covers both that reward uncertainty and transition uncertainty. The theorem uses the coefficient values in equation 36; it makes no calibration claim for arbitrary numerical choices of the bonuses. The lower bound of equation 66 is the only constraint on β_P, namely $\beta_P \ge 4C_N + Hr_{\max}c_{P,N}$; the upper bound uses β_P linearly, giving $a_N = (4 + 2\sqrt{2})C_N + \beta_P$ and $D_N = a_N + Hr_{\max}c_{P,N}$ in equation 65. Any larger β_P is therefore admissible, with C_N replaced by β_P in the leading factor of the bound; the $\widetilde{O}$ statement is unchanged when β_P is a constant multiple $c \ge 1$ of the prescribed value. The argument does not depend on β_B beyond $\beta_B \ge -1$, so the theorem holds for every fixed $\beta_B \in [-1, 1]$.

C.14 Scope of the assumptions and the rate

Identifiability concerns the aggregate design. Take $\mathbf{x}_i = (1,1)^\top$ for every feedback segment and noise identically zero. The reward tables $(1,-1)^\top$ and $(-1,1)^\top$ produce identical zero labels. Both coordinate counts increase, but the ridge estimate stays zero. Moreover, along $v = (1,-1)^\top$, $v^\top A_k v = 2\lambda$, whereas $v^\top \mathrm{diag}(\lambda + N_k(j))v$ grows with the number of segments. No fixed positive η satisfies equation 38. This example rules out coordinatewise learning from noise assumptions alone. It is not an impossibility result for policy learning when the unidentified reward directions are irrelevant to all policy values.

Theorem 1 does not assume an empirical Gram lower bound. Instead, Lemma 1 derives it from a uniform holding probability and a deterministic stationary mapping within each block. These restrictions are material: a stage-dependent policy can choose one action at the first stage and another at the second stage, giving identical action totals in every trajectory even under fully supported state transitions. The proof allows block lengths and segment boundaries chosen from past data. A stopping rule depending on the current trajectory is outside the established design lemma.

Noise, dimensions, and horizon factors. For fixed positive regularization parameters and fixed $S, A, H, p, \sigma, L_{\mathrm{seg}}, N_0$, C_N grows at most as $\sqrt{\log(T+1)}$ for fixed δ. Its explicit definition retains the factors $\sqrt{d}$, $\sqrt{L_{\mathrm{seg}}}$, $\eta^{-1/2}$, and the feature and regularization factor $\sqrt{Q}$; through b_D, the factor $\eta^{-1/2}$ grows like $\sqrt{N_0}$, so a larger initialization set enlarges the certified constant, an artifact of Lemma 1 not exploiting the initialization design. If an aggregate label sums conditionally independent primitive noises with sub-Gaussian scale σ_{step}, its scalar scale can be as large as $\sigma_{\mathrm{step}}\sqrt{H}$; this dependence enters through σ. More generally, correlated label vectors must satisfy the joint bound in equation 13 with their actual scale. The square-root dependence on the interaction count therefore does not imply a horizon-independent prefactor.

Adaptive weights and misspecification. The proof does not bound individual surrogate biases or require surrogate consistency. Misspecification does not enter the leverage comparison, and the sandwich estimator uses reference residuals. There is no probability union bound over the number of surrogates: their contribution to this upper bound is through Q. The result guarantees a regret rate, not a strict improvement over the reference-only method or oracle-optimal MSE for fitted weights.

Planning accuracy and the comparator. The policy class in the regret benchmark must match the one permitted to the learner. A finite-horizon problem can have a strictly better stage-dependent policy than every stationary mapping, even when Assumption 2 holds. For example, take two states $0, 1$, two actions a, b, horizon two, and a uniform initial state. Let

s	$r(s,a)$	$r(s,b)$	$P(1 \mid s,a)$	$P(1 \mid s,b)$
0	0	-0.1	0.1	0.9
1	1	0.9	0.1	0.9

and let the remaining transition probability go to state zero. All self-loop probabilities are at least 0.1. Choosing b at stage one and a at stage two gives return 1.3. The four deterministic stationary mappings have returns $0.6, 0.9, 0.9, 1.2$, so their best return is 1.2. Thus an unrestricted finite-horizon benchmark would add a gap of at least $0.1K$ in this example, regardless of estimation accuracy.

For the stationary comparator, Assumption 3 controls the existing planner's global objective gap relative to Π_{stat}. Its contribution is at most $2c_{\mathrm{opt}}\sqrt{N}$ by equation 74, including when block lengths vary predictably. Exact stationary maximization permits $c_{\mathrm{opt}} = 0$. This is an assumed computational accuracy guarantee, not a consequence of the statistical assumptions. Ordinary finite-horizon value iteration optimizes over stage-dependent policies and does not by itself establish this guarantee for a stationary mapping extracted from its output.

Implementation scope. The linear and fixed-threshold surrogates fall within the theorem's fitting specification. Neural-network fits, paired bootstrap covariances, and general influence-function estimators require their own stability analysis to derive a counterpart of equation 59. The guarantee uses the stated bonus coefficients, one stationary mapping per block, and feedback completion between blocks. No performance claim for the final bonus-free policy follows solely from cumulative collection-policy regret.

Relation to the experiments. None of the reported experiments runs the idealized procedure exactly. The Advertiser MDP satisfies the holding condition with $p = 0.4$, but it is a continuing task: block boundaries do not reset the environment, the planner optimizes average reward by relative value iteration, the bonus coefficients are selected empirically rather than set to equation 36 (with the count bonus scaled by $r_{\max}$), the models are refit after every ten-block batch rather than after every block, the transition estimate uses additive smoothing, and the covariance is the fixed-design expression with the known noise variance rather than the clustered sandwich estimator (the former

satisfies the analogue of equation 59 directly, since $\sigma^2\|a_{m,k,j}\|_2^2 \le Q_m\sigma^2(A_k^{-1})_{jj}$). In the Sepsis simulator the treatment indicators in the state are set by the action, so $P(s \mid s, a) = 0$ for every action that changes a treatment and Assumption 2 fails; episodes end at trajectory-dependent death or discharge times; and the label is additive in successor-dependent rewards $r(s, a, s')$ rather than of the state–action form equation 1, so Assumption 1 fails as well and the fitted table is a pseudo-reward. FSRS-Exam is a continuous-state task outside the tabular analysis. In every experiment the evaluated policy is the final bonus-free policy of Algorithms 1 and 2. The theorem therefore certifies the mixing mechanism under its stated assumptions; it does not by itself predict the reported numbers.

D Complete Algorithms

The complete procedures for the tabular MDP and its continuous-state extension are provided in Algorithms 1 and 2, respectively.

Algorithm 1 Tabular reward learning with value iteration

Require: Initial dataset $\mathcal{D}_0$, surrogate models $\{\widehat{r}_m\}_{m=1}^M$, ridge penalty λ, exploration coefficients β_r, β_P, and collection rounds K

$\mathcal{D} \leftarrow \mathcal{D}_0$
for $k = 1, \dots, K$ **do**
  Construct the count matrix $\mathbf{X}$ and aggregate-label vector $\mathbf{y}$ from $\mathcal{D}$
  Fit the ridge reference $\widetilde{\mathbf{r}} = (\mathbf{X}^\top\mathbf{X} + \lambda I_d)^{-1}\mathbf{X}^\top\mathbf{y}$
  Refit the data-dependent surrogates and evaluate all fixed surrogates
  Estimate the joint covariance $\widehat{\boldsymbol{\Sigma}}_j$ for every state–action pair j
  for each state–action pair j **do**
    Construct the bias proxies $\widehat{\mathbf{b}}_j$ using Equation equation 10
    Form $\widehat{\mathbf{R}}_j = \widehat{\boldsymbol{\Sigma}}_j + \widehat{\mathbf{b}}_j\widehat{\mathbf{b}}_j^\top$
    Solve Equation equation 11 for $\widehat{\mathbf{w}}_j$
    Set $\widehat{r}_{\mathrm{mix},j} = \widehat{\mathbf{w}}_j^\top\widehat{\mathbf{r}}_j$
    Set $b_{r,j} = (\widehat{\mathbf{w}}_j^\top\widehat{\mathbf{R}}_j\widehat{\mathbf{w}}_j)^{1/2}$
  end for
  Estimate $\widehat{P}$ and $N(s, a)$ from all observed transitions
  Set $b_P(s, a) = 1/\sqrt{N(s, a) + 1}$
  Form $r_{\mathrm{plan}} = \widehat{r}_{\mathrm{mix}} + \beta_r b_r + \beta_P b_P$
  $\pi_k \leftarrow$ ValueIteration$(\widehat{P}, r_{\mathrm{plan}})$
  Execute π_k and append the resulting trajectories, transitions, and aggregate labels to $\mathcal{D}$
end for
Refit all models and weights using the complete dataset
$\pi_{\mathrm{final}} \leftarrow$ ValueIteration$(\widehat{P}, \widehat{r}_{\mathrm{mix}})$
return π_{final}

Algorithm 2 Continuous TrustMix with a bootstrapped DNN reference and SAC

Require: Initial labeled trajectories $\mathcal{D}_0$; surrogate reward models $\{\widehat{r}_m\}_{m=1}^{M}$; reward-bonus coefficient β_r; number of bootstrap replicates B_{boot}; number of data-collection rounds K.

Set $\mathcal{D} \leftarrow \mathcal{D}_0$ and initialize matched collection and evaluation SAC agents.
for $k = 0, \ldots, K$ **do**
  Fit a central DNN reward $\widehat{r}_{0,k}$ by minimizing

$$\sum_i \left[y_i - \sum_{t \in \mathcal{B}_i} \widehat{r}_{0,k}(s_t, a_t) \right]^2 + \lambda_1 \sum_{i,t} \left| \widehat{r}_{0,k}(s_t, a_t) \right|.$$

  Fit every data-dependent surrogate using $\mathcal{D}$; leave externally supplied surrogates fixed.
  for $b = 1, \ldots, B_{\mathrm{boot}}$ **do**
    Resample complete trajectories with replacement, preserving all feedback segments within each selected trajectory.
    Independently initialize and train a complete DNN $\widehat{r}_{0,k}^{(b)}$ on the resampled episodes using the same aggregate loss as the central reference.
    Refit every data-dependent surrogate on the same resampled trajectories; keep externally supplied surrogates fixed.
  end for
  for each replay pair $x = (s, a)$ **do**
    Action-center the reference and its bootstrap predictions: $\widehat{r}^{c}_{0,k}(s, a) = \widehat{r}_{0,k}(s, a) - \widehat{r}_{0,k}(s, \mathbf{0})$.
    Evaluate the fixed surrogates and estimate reference uncertainty from the action-centered bootstrap predictions.
    Compute the bias proxies $\widehat{\mathbf{b}}_k(x)$ from the reference–surrogate disagreements, using the declared continuous reference-bias allowance $B_{\phi,k}(x)$.
    Form $\widehat{\mathbf{R}}_k(x)$, solve Equation equation 11, and set

$$\widehat{r}_{\mathrm{mix},k}(s, a) = \widehat{r}_{0,k}(s, \mathbf{0}) + \widehat{w}_{0,k}\widehat{r}^{c}_{0,k}(s, a) + \sum_{m=1}^{M} \widehat{w}_{m,k}\widehat{r}_m(s, a).$$

    Compute $b_{r,k}(s, a)$ from the weighted reference uncertainty and the bias proxies.
  end for
  Rebuild the collection replay buffer from all complete episodes and relabel it using

$$r_{\mathrm{plan},k}(s, a) = \widehat{r}_{\mathrm{mix},k}(s, a) + \beta_r b_{r,k}(s, a).$$

  Update the collection SAC actor and critics using the relabeled buffer and SAC's entropy regularization.
  Rebuild a matched evaluation buffer using $\widehat{r}_{\mathrm{mix},k}$ without the bonus and update the evaluation SAC agent.
  if $k < K$ **then**
    Execute the collection policy, obtain new complete episodes and aggregate labels, and append them to $\mathcal{D}$.
  end if
end for
return the bonus-free evaluation policy π_{final}.

E Detailed Experimental Setup

E.1 Advertiser MDP

Environment. We consider a continuing MDP in which an advertiser observes daily shopper traffic in a mall and chooses among low, medium, and high advertising intensities. The state space is $\mathcal{S} = \{0, \ldots, 9\}$, the action space is $\mathcal{A} = \{0, 1, 2\}$, and every run starts from $s_0 = 0$. For an interior state, the transition probabilities are

Action	$P(s_{t+1} = s_t - 1)$	$P(s_{t+1} = s_t)$	$P(s_{t+1} = s_t + 1)$
Low ($a = 0$)	0.6	0.4	0
Medium ($a = 1$)	0.3	0.4	0.3
High ($a = 2$)	0	0.4	0.6

At the boundary states $s = 0$ and $s = 9$, probability mass that would leave the state space is added to the self-transition probability. Feedback-block boundaries do not reset the environment.

The latent primitive reward is

$$r^\star(s,a) = 0.05s - 0.005s^2 - 0.05a - 0.12a^2 + \eta_a s, \qquad (\eta_0, \eta_1, \eta_2) = (0, 0.02, 0.05). \tag{76}$$

The realized reward table lies in $[-0.58, 0.125]$, and the algorithms use the known absolute bound $\|r^\star\|_\infty \le r_{\max} = 0.6$.

The learner observes every state, action, and next state, but not the primitive rewards. After each monthly block $\mathcal{B}_i$ of $T = 30$ transitions, interpreted as one monthly report, it observes only the noisy aggregate label

$$y_i = \sum_{t \in \mathcal{B}_i} r^\star(s_t, a_t) + \varepsilon_i, \qquad \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.2^2). \tag{77}$$

Thus, the learner-facing record for a block contains its transitions, state–action counts, and y_i, whereas the latent primitive rewards are retained separately and used only for evaluation.

We optimize the continuing average reward

$$g(\pi) := \liminf_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mu_0, P, \pi} \left[\sum_{t=0}^{N-1} r^\star(s_t, a_t) \right]. \tag{78}$$

Solving the true MDP gives the optimal deterministic policy

$$\pi^\star(s) = \begin{cases} 1, & s \in \{0, 6, 7, 8, 9\}, \\ 2, & s \in \{1, 2, 3, 4, 5\}. \end{cases}$$

Hence, high advertising is optimal while traffic is being built, whereas medium advertising is sufficient at high-traffic states.

Reward models. Let $c_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ denote the vector of state–action visit counts in block i. For a feature table Φ_m, reward model m is fitted directly to the aggregate labels by ridge regression:

$$\widehat{\theta}_m = \arg\min_\theta \sum_i \left(y_i - c_i^\top \Phi_m \theta \right)^2 + \lambda \|\theta\|_2^2, \qquad \lambda = 10^{-3}. \tag{79}$$

The resulting primitive-reward prediction is $\widehat{r}_m(s,a) = \phi_m(s,a)^\top \widehat{\theta}_m$.

The reference model is a 30-parameter tabular model, with one parameter for every state–action pair. TrustMix additionally uses two lower-dimensional surrogates fitted to the same aggregate labels:

1. A four-parameter bilinear model,

$$\widehat{r}_{\text{lin}}(s,a) = \theta_0 + \theta_1 s + \theta_2 a + \theta_3 sa. \tag{80}$$

2. A six-parameter threshold model, with one action-specific constant in each of the two regions $s \le 4$ and $s \ge 5$.

Because all three estimators are fitted to the same labels, their estimation errors are correlated. TrustMix therefore uses the full joint fixed-design covariance, including cross-model terms, with the known innovation variance 0.2^2. At each state–action pair, it combines the three predictions using simplex-constrained weights that minimize the local proxy error matrix, which contains both covariance and squared-bias terms.

Both surrogate classes are deliberately coarse and misspecified. The bilinear model omits the quadratic terms in Equation equation 76, whereas the threshold model discards all within-region traffic variation. They represent plausible but imperfect domain models rather than oracle approximations to the true reward.

Baselines and fixed-weight ablation. We compare TrustMix with the following baselines.

- **Delayed Differential Q-learning.** Once a block label arrives, the method assigns zero pseudo-reward to the first 29 transitions and assigns the complete aggregate label y_i to the final transition. It then replays the block once in chronological order using undiscounted Differential-Q updates. The reported policy is greedy with respect to the learned action-value table. We use Q step size 0.01, gain ratio 0.01, and constant ϵ-greedy exploration with $\epsilon = 0.05$. The aggregate label is not divided by the block length.
- **Aggregate REINFORCE.** This baseline uses a tabular softmax policy and holds the policy fixed for a window of ten blocks. It centers each block label using the running estimate of average reward per primitive step and forms reverse cumulative block-level returns-to-go within the window. One on-policy gradient update is then applied with learning rate 3.0. No gradient or logit clipping is used. Evaluation uses the learned stochastic softmax policy rather than its greedy transformation.
- **Reference-only.** This baseline fits only the 30-parameter tabular reward model. It uses the same transition estimator and average-reward planner as TrustMix, but has no surrogate models to combine. Consequently, its reward-model weight is identically one on the tabular reference.

The cumulative-regret figure additionally includes **Uniform Weight**, a fixed-weight ablation that assigns weight $(1/3, 1/3, 1/3)$ to the tabular, bilinear, and threshold reward models at every state–action pair. It otherwise uses the same reward fits, covariance and bias calculations, transition estimator, exploration bonuses, and planner as TrustMix.

Model-based planning. Reference-only, TrustMix, and Uniform Weight estimate the transition kernel using additive smoothing 10^{-3}. They refit their reward and transition models after each ten-block collection batch. During data collection, the planning reward is

$$\widehat{r}_{\mathrm{mix}}(s,a) + \beta_r b_r(s,a) + \beta_P \frac{r_{\max}}{\sqrt{N(s,a)+1}}, \tag{81}$$

where $b_r(s,a)$ is the reward-model uncertainty term and $N(s,a)$ is the cumulative visit count. The selected coefficients are

$$(\beta_r, \beta_P) = \begin{cases} (0.1, 0.25), & \text{Reference-only,} \\ (0.375, 0.175), & \text{TrustMix and Uniform Weight.} \end{cases}$$

Policies are computed using relative value iteration with tolerance 10^{-10} and at most $20{,}000$ iterations, with an exact average-reward policy-iteration fallback for slowly mixing smoothed kernels. Exploration bonuses are removed when constructing every reported evaluation policy.

E.2 Sepsis Simulator

Environment. We use a finite-horizon episodic sepsis-treatment MDP adapted from the synthetic simulator of Oberst and Sontag (Oberst & Sontag, 2019). The simulator is mechanistic and is not calibrated for clinical decision-making. A physiological state records diabetes status, heart rate, blood pressure, oxygen saturation, glucose level, and indicators for whether antibiotics, vasopressors, and mechanical ventilation are currently active. The corresponding cardinalities are

$$(2, 3, 3, 2, 5, 2, 2, 2),$$

giving $1{,}440$ physiological configurations. We add separate absorbing death and discharge states, so the complete state space has

$$|\mathcal{S}| = 1{,}442.$$

The action specifies a binary choice for each of the three treatments, giving $|\mathcal{A}| = 2^3 = 8$ possible treatment combinations. We preserve the original hand-specified physiological transition dynamics and reset distribution, with nominal diabetes probability 0.2 and all treatments initially inactive.

Each patient episode contains at most $H = 20$ treatment decisions and ends earlier upon death or discharge. Thus, if τ denotes the episode length,

$$\tau = \min\{\text{first death or discharge time},\ H\}.$$

An episode that reaches H without either outcome is truncated.

Stochastic mortality and aggregate feedback. The original simulator uses deterministic terminal conditions. We retain its discharge rule but replace its deterministic demise rule with a synthetic stochastic mortality mechanism. After an action is applied, the simulator first draws the next physiological configuration x' using the original transition dynamics. If all four physiological variables are normal and all treatments are inactive in x', the patient is discharged. Otherwise, death occurs with probability

$$\begin{aligned} p_{\mathrm{death}}(x') = p_0(k(x')) &+ 0.05\,\mathbf{1}\{\mathrm{BP}(x') \text{ is low}\} \\ &+ 0.05\,\mathbf{1}\{\mathrm{O}_2(x') \text{ is low}\} \\ &+ 0.05\,\mathbf{1}\{\mathrm{diabetic}(x')\}\mathbf{1}\{\mathrm{glucose}(x') \text{ is abnormal}\}, \end{aligned} \tag{82}$$

where $k(x') \in \{0, \dots, 4\}$ is the number of abnormal heart-rate, blood-pressure, oxygen, and glucose indicators, and

$$\big(p_0(0), p_0(1), p_0(2), p_0(3), p_0(4)\big) = (0.01, 0.03, 0.10, 0.65, 0.80).$$

These coefficients are chosen solely to construct a stochastic benchmark and should not be interpreted as clinical mortality estimates.

The transition reward is

$$r(s, a, s') = \begin{cases} +1, & s' = s_{\mathrm{discharge}}, \\ -1, & s' = s_{\mathrm{death}}, \\ 0, & \text{otherwise.} \end{cases} \tag{83}$$

The learner observes the complete state–action trajectory but receives only the aggregate episode label

$$y_i = \sum_{h=0}^{\tau_i - 1} r(s_{i,h}, a_{i,h}, s_{i,h+1}) \in \{-1, 0, +1\}. \tag{84}$$

No additional Gaussian label noise is added. Because the outcome depends on the realized successor state, the fitted state–action reward tables should be interpreted as policy-useful pseudo-reward decompositions rather than as an identified latent clinical reward.

Reward models. The reference model is a tabular ridge regression over the complete state–action domain, with $1{,}442 \times 8 = 11{,}536$ coordinates. TrustMix combines this reference with two learned surrogate models fitted to the same episode-level labels.

The *physiology surrogate* contains an intercept and categorical main effects for the eight state variables, crossed with the selected action. It has 14 state features per action and therefore 112 coefficients. The *severity surrogate* groups states by diabetes status and the number of abnormal physiological indicators, and assigns a separate value to every group–action combination. It has $2 \times 5 \times 8 = 80$ coefficients.

Full TrustMix additionally includes a fixed doctor-prior score. This prior assigns score -1 when the current physiological state has at least three abnormal indicators, score $+1$ when all four indicators are normal and the selected action turns all treatments off, and score 0 otherwise. It is a handcrafted auxiliary signal: it is not fitted to labels and has no access to the transition kernel or the mortality probabilities in Equation equation 82. The no-doctor ablation removes only this fixed prior while retaining both learned surrogates and all TrustMix hyperparameters.

Baselines. We compare against four baselines. Delayed Q-learning is a time-indexed tabular adaptation of Q-learning (Watkins & Dayan, 1992): after an episode ends, the observed label is assigned to its final transition and all preceding transitions receive zero reward. Aggregate REINFORCE is a tabular softmax policy-gradient method based on REINFORCE (Williams, 1992), in which the episode label weights every action in the completed trajectory. UCBVI-TS is our episodic implementation of the trajectory-feedback method of Efroni et al. (Efroni et al., 2021); it combines a tabular aggregate-label reward estimate, one correlated Gaussian reward perturbation per episode, and transition optimism. Reference-only uses the same model-based planning framework as TrustMix but excludes all surrogate reward models. We also report TrustMix without the fixed doctor prior as a structural ablation.

E.3 FSRS-EXAM: ENVIRONMENT AND IMPLEMENTATION DETAILS

Environment. FSRS-EXAM is a finite-horizon episodic MDP with horizon $H = 20$ and ten exchangeable topics. At decision time $t \in \{0, \dots, H-1\}$, topic k is described by its FSRS memory stability $S_{t,k}$, difficulty $D_{t,k}$, elapsed days $E_{t,k}$ since its most recent review, review count $N_{t,k}$, and lapse count $L_{t,k}$. Stability is measured in days: under the FSRS forgetting curve, $S_{t,k}$ is the delay at which recall probability decreases to 0.9. The learner-visible state is

$$s_t = \left(\frac{t}{H}, \left\{ \log(1 + S_{t,k}),\ \frac{D_{t,k}}{10},\ \frac{E_{t,k}}{H},\ \frac{N_{t,k}}{H},\ \frac{L_{t,k}}{H} \right\}_{k=1}^{10} \right) \in \mathbb{R}^{51}. \tag{85}$$

All five coordinates are set to zero for an unseen topic.

The continuous action $a_t \in [-1, 1]^{10}$ assigns a logit to each topic. The environment samples the topic reviewed on that day from

$$p_k(a_t) = \frac{\exp(2a_{t,k})}{\sum_{j=1}^{10} \exp(2a_{t,j})}. \tag{86}$$

Thus, SAC controls a distribution over topics rather than directly choosing a discrete topic.

Each topic evolves independently according to FSRS-6 when reviewed (Open Spaced Repetition, 2026b). For a previously studied topic with stability S, its probability of successful recall after Δ days is

$$R(\Delta, S) = \left(1 + F\frac{\Delta}{S}\right)^{d}, \qquad F = 0.9^{1/d} - 1, \qquad d < 0, \tag{87}$$

so that $R(S, S) = 0.9$. The decay d and the coefficients governing difficulty and stability updates are determined by the student's 21-dimensional FSRS-6 parameter vector. A failed recall produces an *Again* rating, while a successful recall produces *Hard*, *Good*, or *Easy*; the resulting rating updates $D_{t,k}$ and $S_{t,k}$ through the FSRS-6 recurrences. For an unseen topic, the first rating initializes its difficulty and stability. Topics not reviewed on that day retain their memory parameters while their elapsed times increase.

After the twentieth study action, every topic is tested once on day 21. Conditional on the final memory states, the ten test outcomes $Z_1, \dots, Z_{10}$ are independent Bernoulli draws whose success probabilities are given by equation 87. The agent observes only the average exam score

$$Y = \frac{1}{10}\sum_{k=1}^{10} Z_k = \frac{1}{10}\sum_{k=1}^{10} \mathbf{1}\{\text{topic } k \text{ is recalled}\} \in \{0, 0.1, \dots, 1\}. \tag{88}$$

Equivalently, the environment supplies zero reward during the first 19 transitions and Y on the final transition. Neither a daily reward nor the exam recall probabilities are observed. Consequently, any learned daily reward is a policy-oriented pseudo-reward rather than an identified primitive reward. For evaluator-only diagnostics, we define the oracle potential as the mean day-21 recall probability under the current memory states and its one-step increment as the oracle pseudo-reward. Neither oracle quantity is available to any agent.

Real-world calibration. The transition coefficients are not manually selected. We use 54 real-user profiles from the FSRS benchmark: 50 profiles are reserved for the 50 evaluation seeds and four disjoint profiles are used for model and hyperparameter selection. Each profile contains 21 FSRS-6 parameters fitted to that user's Anki review history (Ye et al., 2022; Su et al., 2023; Open Spaced Repetition, 2026c). The probabilities of the four initial ratings and the conditional Hard/Good/Easy probabilities after a successful recall are estimated from the same user's empirical answer frequencies (Open Spaced Repetition, 2026a). An evaluation seed is paired with one fixed held-out profile. Hence the 20-day curriculum and final exam are designed, but forgetting, recall, and post-review memory updates are governed by parameters estimated from real behavior.

LLM queries. We generated candidate states from 50 random-action and 50 round-robin trajectories using one of the four tuning profiles. We then selected 100 policy-and-time-stratified slots from each source policy. One exact duplicate across policies reduced the 200 slots to 199 distinct

query states; its annotation was reused with sampling weights that preserved the balanced design. No held-out evaluation profile was used to construct the query set.

We queried `claude-opus-5-5`, `claude-sonnet-5`, `gpt-5.6-sol`, and `gpt-6-astra` independently at every selected learner state in September 2026; the two OpenAI models used medium reasoning effort. The model received only the learner-visible state variables, decoded into their natural units; it was not given the FSRS equations, fitted user parameters, transition samples, exam outcomes, or oracle recall probabilities. Topic aliases and their presentation order were independently randomized for every query. We used the same system prompt for all four models:

```
You are an annotation model providing a frozen domain prior for a study-scheduling experiment.

<context>
A learner studies 10 anonymous, exchangeable, content-neutral knowledge concepts. There are 20
     consecutive study days, and exactly one concept is reviewed each day. The learner state
    is observed immediately before today's review. On day 21, every concept is tested exactly
     once, all 10 test items have equal weight, and the objective is the mean probability of
    correct recall. The same unknown learner-response process applies to every concept.
</context>

<task>
The learner must choose exactly one concept to review today. For each possible choice,
    estimate how much reviewing that concept today would increase the expected mean score on
    the day-21 exam, relative to not reviewing that concept today.

Average over uncertainty in today's review outcome. For this local annotation only, assume
    that after today's review no concept is reviewed again before the exam, while memory
    continues to decay naturally. Rank the ten choices from largest to smallest expected
    improvement. Rank the marginal benefit of today's review, not merely current forgetting
    risk.
</task>

<feature_meanings>
- seen: whether the concept has ever been reviewed.
- strength_days: an estimate of memory durability in days. Larger values mean slower
    forgetting. This is not a recall probability.
- difficulty_1_to_10: estimated learning difficulty on a 1-to-10 scale. Larger values mean
    harder.
- elapsed_days: whole days since the most recent review.
- reviews: total number of previous reviews, including the first exposure.
- lapses: number of failed recall attempts after initial exposure.

For an unseen concept, seen is false; strength_days, difficulty_1_to_10, and elapsed_days are
    null; and reviews and lapses are 0.
</feature_meanings>

<rules>
1. Use only the supplied learner state.
2. Concept aliases and their display order are random and carry no information.
3. Concepts have no content-specific differences beyond the supplied features.
4. Priority tier 1 is the highest priority.
5. Ties are allowed and should be used whenever the supplied evidence does not justify an
    ordering.
6. Concepts with identical displayed feature values MUST be placed in the same tier.
7. Never break a tie using concept alias or display order.
8. Include every concept alias exactly once.
9. Return only JSON conforming to the supplied schema.
10. Do not return scores, confidence values, explanations, reasoning, markdown, or additional
    text.
</rules>
```

For each state, the user message instantiated the following template. Continuous state features were transformed back to their natural units and rounded to four decimal places. Concept identifiers and array order were randomized independently:

```
<learner_state>
{
  "state_id": "q_XXXXXX",
  "study_day": <integer from 1 to 20>,
  "days_until_final_test": <integer from 1 to 20>,
  "concepts": [
    {
      "concept_id": "K0",
      "seen": <true or false>,
      "strength_days": <number or null>,
      "difficulty_1_to_10": <number or null>,
      "elapsed_days": <integer or null>,
```

```
    "reviews": <integer>,
    "lapses": <integer>
   },
   ...
  ]
}
</learner_state>

Rank this learner state according to the system criterion.
```

The provider-side structured-output constraint required an object with exactly three fields:

```
{
  "schema_version": "priority-tiers-v1",
  "state_id": "q_XXXXXX",
  "priority_tiers": [
    ["K3"],
    ["K1", "K7"],
    ["K0", "K2", "K4", "K5", "K6", "K8", "K9"]
  ]
}
```

The first inner list is the highest-priority tier. Every alias had to appear exactly once, and concepts with identical displayed features were required to be tied. Invalid responses were rejected and retried.

The counterfactual assumption that no topic is reviewed again after the current action makes each annotation a local marginal-action comparison. It avoids asking the LLM to posit an unspecified continuation policy, while still eliciting domain knowledge about forgetting, spacing, difficulty, and the remaining time until the exam. The resulting rankings were used only to fit the frozen surrogate reward models described below.

From rankings to frozen surrogate rewards. A separate reward network is trained for each LLM. If topic k has midrank ρ_{imk} in model m's tiered ranking at query state s_i, its target is the centered Borda score

$$z_{imk} = \frac{5.5 - \rho_{imk}}{4.5}. \tag{89}$$

Tied topics receive their shared average rank. Thus $z_{imk} \in [-1, 1]$, $\sum_k z_{imk} = 0$, and a completely tied ranking maps to zero for all topics.

The reward network is permutation equivariant across topics. A shared local encoder embeds each topic's memory features together with the current day. Mean and maximum pooling over the ten embeddings produce a global student representation, and a shared scoring network outputs one score per topic. The local encoder has two 64-unit SiLU layers, and the scoring network has one 128-unit SiLU hidden layer. The scores are centered within each state and rescaled only when necessary to obtain

$$g_{mk}(s) \in [-1, 1], \qquad \sum_{k=1}^{10} g_{mk}(s) = 0. \tag{90}$$

The network is trained by mean squared error against equation 89; train/validation splits are made by state and stratified by decision time.

For a continuous SAC action, the frozen surrogate is

$$\widehat{r}_m(s, a) = c_{\mathrm{LLM}} \sum_{k=1}^{10} p_k(a) g_{mk}(s), \tag{91}$$

using the same topic probabilities as the simulator in equation 86. We tune one common $c_{\mathrm{LLM}} \in \{0.0125, 0.025, 0.05\}$ on the four tuning profiles and freeze it before evaluation; the selected value is 0.05. We do not tune separate scales for different LLMs. Equation equation 91 guarantees $|\widehat{r}_m(s, a)| \leq c_{\mathrm{LLM}}$ and prevents arbitrary reward scales from entering SAC. The four surrogate networks are fixed throughout online learning and never use observed exam scores.

Because rankings identify within-state action preferences but not an action-independent reward level, TrustMix combines $\widehat{r}_m(s, a)$ with the action-centered component of the neural reference. The

reference's action-independent component is then added back to the mixed reward. The reference and local TrustMix weights are refitted from accumulated exam feedback, whereas the four LLM surrogates remain unchanged.

Baselines and matched control protocol. *Delayed-reward SAC* assigns zero reward to the first 19 transitions and the observed exam score to the terminal transition. *RUDDER-SAC* trains a causal return-prediction LSTM on complete episodes and uses consecutive prefix-prediction differences, followed by a return-preserving residual correction, as SAC rewards (Arjona-Medina et al., 2019). Before sufficient return diversity is available, it reduces to delayed-reward SAC. *Reference-only SAC* fits a neural per-transition reward whose episode sum predicts the observed exam score; a paired episode-bootstrap ensemble provides its exploration bonus. TrustMix replaces the reference-only reward with the mixture of the neural reference and the four frozen LLM surrogates.

All methods use $\gamma = 1$, identical SAC architectures and initializations, the same warm-up episodes and online data budgets, genuine terminal flags, and complete-episode replay. Whenever a learned reward is updated, all stored transitions are relabeled before SAC training so that stale and current reward versions never coexist.

Implementation details. Each run begins with 20 shared warm-up episodes, adds ten online episodes per round, and refits the learned reward model after every round. The reference is a two-hidden-layer DNN that maps the normalized 51-dimensional state and 10-dimensional action to a scalar, with 128 SiLU units per hidden layer. At each refit, we train the central network and five independently initialized full-network bootstrap members for 100 epochs on episode-level resamples; the loss includes an ℓ_1 penalty of 10^{-4} on per-transition predictions. Thus, the bootstrap refits the entire DNN rather than only its last layer. For TrustMix, the reference-bias allowance is 0.05, equal to the selected LLM reward scale. Replay data are augmented by all ten cyclic permutations of the topic indices. The SAC actor and critics each use two 256-unit hidden layers, a learning rate of 3×10^{-4}, batch size 256, $\tau = 0.005$, fixed entropy coefficient 0.003, and 250 gradient steps per reward update. RUDDER-SAC uses a one-layer 64-unit LSTM trained for 75 epochs. Hyperparameters were selected using only the four tuning profiles and then fixed for all 50 held-out evaluation profiles. Every 20 online episodes, we evaluate the deterministic SAC actor in 64 simulator rollouts and report the mean conditional expected exam score; the exam Bernoulli draws are not used in this evaluation metric.

E.4 Additional Tables and Figures

Table 2: Policy optimality gap at fixed episode budgets. Entries report $V_H^\star(\mu) - V_H^{\widehat{\pi}_k}(\mu)$ in raw episode-return units as the mean $\pm$ pointwise 95% Student-t confidence-interval half-width across 50 matched seeds. Warm-up episodes are included. Lower is better.

Episodes	Q-learning	REINFORCE	UCBVI-TS	Reference-only	TrustMix (w/o doctor prior)	TrustMix
10	0.875 ± 0.001	$0.876 \pm <0.001$	0.832 ± 0.005	0.832 ± 0.005	0.864 ± 0.026	$\mathbf{0.718 \pm 0.027}$
20	0.872 ± 0.002	0.875 ± 0.001	0.824 ± 0.007	0.847 ± 0.009	0.810 ± 0.034	$\mathbf{0.705 \pm 0.023}$
50	0.868 ± 0.002	0.872 ± 0.002	0.802 ± 0.013	0.841 ± 0.011	0.770 ± 0.039	$\mathbf{0.641 \pm 0.020}$
100	0.856 ± 0.004	0.867 ± 0.002	0.809 ± 0.015	0.835 ± 0.010	0.700 ± 0.043	$\mathbf{0.624 \pm 0.019}$
200	0.831 ± 0.008	0.857 ± 0.004	0.806 ± 0.016	0.835 ± 0.012	0.627 ± 0.039	$\mathbf{0.606 \pm 0.015}$
400	0.798 ± 0.008	0.840 ± 0.005	0.791 ± 0.021	0.824 ± 0.012	$\mathbf{0.569 \pm 0.041}$	0.583 ± 0.014
600	0.778 ± 0.008	0.825 ± 0.006	0.767 ± 0.022	0.823 ± 0.012	$\mathbf{0.532 \pm 0.033}$	0.580 ± 0.016

Here $V_H^\pi(\mu) := \mathbb{E}_{\mu,P,\pi}[\sum_{h=0}^{H-1} r_h]$ with $H = 20$. A positive half-width that would round to 0.000 is reported as <0.001. Evaluation uses the exact bonus-free policy value. Episode lengths vary, so equal episode budgets need not imply equal transition counts. Bold denotes the lowest sample mean among methods shown, not statistical significance. The no-doctor ablation inherits the TrustMix hyperparameters; UCBVI-TS denotes the implemented adaptation.

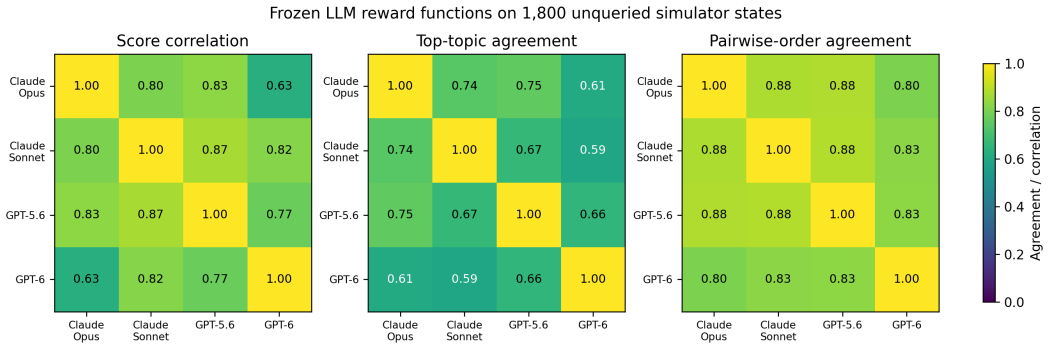


Figure 4: **Agreement among the four frozen LLM reward models.** Pairwise score correlation, top-topic agreement, and pairwise-order agreement are evaluated on 1,800 unqueried simulator states. The models exhibit related but nonidentical study preferences.

Table 3: **FSRS-EXAM results.** Values are means ± standard errors across 50 paired held-out user profiles. The final column reports the paired TrustMix-minus-comparator difference with a 95% confidence interval. Higher is better.

Method	Final expected exam score	Normalized AUC	Final paired TrustMix gain
TrustMix-SAC	$\mathbf{0.8358 \pm 0.0188}$	$\mathbf{0.8204 \pm 0.0187}$	—
Delayed SAC	0.8077 ± 0.0196	0.7878 ± 0.0189	$0.0281\ [0.0241,\ 0.0321]$
Reference-only SAC	0.8015 ± 0.0191	0.7654 ± 0.0175	$0.0343\ [0.0309,\ 0.0378]$
RUDDER-SAC	0.7777 ± 0.0191	0.7562 ± 0.0182	$0.0581\ [0.0528,\ 0.0635]$
Uniform random	0.7386 ± 0.0150	—	$0.0972\ [0.0891,\ 0.1053]$
Round-robin	0.8322 ± 0.0173	—	$0.0037\ [-0.0005,\ 0.0078]$

Normalized AUC is the trapezoidal area under the expected-score learning curve from episodes 20 to 500, divided by the interval length; it therefore has the same $[0, 1]$ scale as the exam score. Round-robin cyclically assigns the highest action logit to topics $1, \ldots, 10$ in order, repeating the cycle twice over 20 days, while ignoring the learner state. The selected topic is still sampled from the resulting softmax distribution. AUC is omitted for the two fixed, non-learning heuristic policies.