WHAT TO PRACTICE NEXT? LATENT PROFILE-GUIDED TASK SELECTION FOR SKILL-EVOLVING LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Task selection shapes the experiences available for skill evolution in LLM agents, yet fixed practice schedules overlook differences in skill requirements and evolving task value. Most existing methods, however, rely on fixed schedules or scalar measures of progress, leaving nonstationary skill demands unaddressed. We propose SPG-Select, a framework for allocating limited practice budget using a multidimensional profile inferred from binary outcomes via MIRT. The framework predicts surrogate progress, defined as the difference between consecutive profile estimates, and prioritizes progress in dimensions with larger estimated proficiency gaps. Our SPG-Select-R policy combines these predictions with uncertainty-based exploration and periodic restarting to accommodate nonstationarity. We evaluate SPG-Select on ALFWorld and WebShop with three skill-evolution backends. In the reported runs, SPG-Select-R improves held-out terminal success over uniform sampling across all six backend-benchmark combinations, with gains of 6.7-15.7 percentage points on ALFWorld and 1.6-10.0 points on WebShop. Under a predictable linear surrogate model and boundedness assumptions, we establish a high-probability dynamic pseudo-regret bound for surrogate task allocation in terms of parameter path variation.

1 INTRODUCTION

Large language model (LLM) agents can improve through repeated interaction, reflection, and reuse of accumulated experience (Yao et al., 2022b; Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2023). Recent work further frames self-evolution as a lifelong process coupling experience exploration, persistent memory, and reusable skill learning (Cai et al., 2025; Wu et al., 2025; Xia et al., 2026). In this work, skill evolution refers to extracting reusable knowledge from trajectories for subsequent decisions. Since the quality of skill evolution depends on which experiences are acquired as well as how they are summarized and retrieved, practice-task selection is a central design decision.

With a limited interaction budget, maximizing immediate success can overallocate practice to already-mastered tasks, whereas more challenging tasks may better develop weak abilities even when they initially fail. Different tasks can also affect skill dimensions unevenly. This motivates ranking tasks by expected surrogate progress relative to the current profile estimate rather than by immediate success or task difficulty alone.

The policy observes only binary outcomes, which reveal neither the contributing latent dimensions nor the true skill change after a backend update. We therefore infer a multidimensional profile as an estimate of the latent skill state and model temporal differences between consecutive profile estimates online. Because accumulated knowledge can change subsequent outcomes and hence the distribution of these temporal differences, older observations can become stale. Task selection is consequently an allocation problem with indirect, nonstationary feedback rather than a static ranking problem or a conventional contextual bandit problem with directly observed scalar rewards.

Curriculum learning has shown that task order can affect optimization and generalization (Bengio et al., 2009), while automatic curricula adapt task distributions to evolving capabilities (Portelas et al., 2020). Self-Evolving Curriculum uses a nonstationary bandit over problem categories during LLM reinforcement-learning fine-tuning (Chen et al., 2025), while Experience-Driven Lifelong

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.

Learning couples experience exploration, skill abstraction, and memory evolution (Cai et al., 2025). We instead study finite-budget task selection from a fixed heterogeneous task pool, using a profile inferred from binary outcomes as an estimate of the latent skill state.

We treat each interaction as both an assessment that updates the profile estimate and an intervention that may affect subsequent behavior through backend updates. Online multidimensional item response theory (MIRT) produces this latent profile estimate rather than a directly observed skill vector, while a contextual bandit selects practice tasks using predicted dimension-wise surrogate progress, i.e., temporal differences between consecutive profile estimates, together with uncertainty. Section 3 explains why MIRT is structurally suited to this role. Its Bernoulli observation model matches the available binary feedback, it represents task difficulty separately from profile-conditioned proficiency, and its vector-valued profile enables the dimension-wise gap weighting used by the policy. The policy thereby targets profile-conditioned surrogate progress rather than task difficulty or success probability alone. We call this framework Skill Profile-Guided Task Selection (SPG-Select). We model task selection as a nonstationary contextual bandit, allowing the conditional mean of surrogate progress to vary across rounds. To limit stale observations, we instantiate SPG-Select with a sliding-window upper confidence bound (UCB) (Cheung et al., 2019) as SPG-Select-W and with periodic restarting (Zhao et al., 2020) as SPG-Select-R. We examine whether this decision layer applies across a skill bank, an insight bank, and shared reflection, answering the question: *can profile-guided task selection improve the use of a limited practice budget?*

To address this question, we make four contributions. First, we formulate finite-budget task selection from a fixed pool as a nonstationary contextual bandit. Temporal differences between MIRT profile estimates provide vector-valued surrogate feedback, which does not directly measure true skill change. A profile-gap-weighted scalar reward emphasizes surrogate progress in dimensions with lower estimated proficiency. Second, we develop SPG-Select, which combines MIRT profile estimation, a task-profile encoder, vector-valued surrogate-progress prediction, and gap-weighted UCB selection under nonstationarity. Third, we evaluate the selector with SkillRL, Experiential Learning (ExpeL), and Shared Reflexion on ALFWorld and WebShop. In the reported runs, SPG-Select-R improves held-out success over Uniform in all six backend-benchmark combinations. Component ablations assess the uncertainty bonus and profile-gap weighting. Fourth, under a predictable time-varying linear surrogate model and boundedness assumptions, we establish a high-probability dynamic pseudo-regret bound for SPG-Select-R in terms of parameter path variation and characterize the oracle restart-period tradeoff. The guarantee concerns surrogate task allocation.

Related work. Our study connects experience-based skill evolution in LLM agents, including linguistic self-feedback, transferable insight extraction, and reusable skill learning (Shinn et al., 2023; Zhao et al., 2024; Xia et al., 2026), with multidimensional ability estimation through MIRT and adaptive assessment (Reckase, 2006; Sharpnack et al., 2024). Experience-based methods provide mechanisms for evolving knowledge from selected practice trajectories, while MIRT provides a latent-state estimate, denoted by the profile, for interpreting binary outcomes. We use this profile for task allocation by drawing on contextual and nonstationary bandits, including linear UCB, sliding-window estimation, and periodic restarting to address parameter drift (Abbasi-Yadkori et al., 2011; Cheung et al., 2019; Zhao et al., 2020). Appendix A provides a fuller account of related work.

2 Problem Setup

Consider an LLM agent interacting with a fixed task pool used for evolution $\mathcal{T} = \{\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(M)}\}$ consisting of $M \in \mathbb{N}_+$ tasks over $T \in \mathbb{N}_+$ successive bandit rounds indexed by $t \in [T] = \{1, \ldots, T\}$. All tasks in $\mathcal{T}$ remain available at every round. Immediately before the decision at round t, the agent has a true latent skill state $\mathbf{s}_t^\star \in [0,1]^K$. This state is assumed to exist, represents the agent's proficiency in $K \in \mathbb{N}_+$ skill dimensions, and cannot be directly observed. We use the term profile exclusively for the MIRT estimate of this latent state. Specifically, the pre-decision profile is the predictive mean $\widehat{\mathbf{s}}_t \in [0,1]^K$ inferred from all observations available before round t. The agent selects a task $\tau_t \in \mathcal{T}$, executes it, and observes a binary environmental outcome $y_t \in \{0,1\}$, where 1 denotes success and 0 denotes failure. Following this interaction and the ensuing skill-evolution update, the unobserved latent state becomes $\mathbf{s}_{t+1}^\star$, giving the true skill change $\boldsymbol{\Delta}_t^\star = \mathbf{s}_{t+1}^\star - \mathbf{s}_t^\star$. Separately, the online MIRT filter uses y_t to produce the next pre-decision profile estimate $\widehat{\mathbf{s}}_{t+1}$. We define the observable bandit feedback, surrogate progress, as the temporal

difference between consecutive profile estimates,

$$\widehat{\boldsymbol{\delta}}_t := \widehat{\mathbf{s}}_{t+1} - \widehat{\mathbf{s}}_t.$$

Thus, $\widehat{\mathbf{s}}_t$ denotes a profile estimate and $\widehat{\boldsymbol{\delta}}_t$ denotes the corresponding temporal difference. The latter is an outcome-based proxy that reflects both belief revision and the state-transition model. It does not directly measure $\boldsymbol{\Delta}_t^\star$ or the effect of the backend update. We use this surrogate to define the practice-allocation objective below.

Each task $\tau \in \mathcal{T}$ is associated with a fixed task embedding $\mathbf{e}_\tau \in \mathbb{R}^{d_e}$, where $d_e \in \mathbb{N}_+$ denotes the embedding dimension. We obtain this embedding by applying a sentence encoder to the task's natural-language description. Before choosing a task at round t, the agent therefore has access to the current profile $\widehat{\mathbf{s}}_t$ and the embedding $\mathbf{e}_\tau$ of every candidate task. We formulate adaptive task selection as a nonstationary contextual bandit problem, with context $(\widehat{\mathbf{s}}_t, \mathbf{e}_\tau)$ for candidate τ. Skill evolution can change the underlying latent state, while successive outcomes update the profile estimate and may alter the expected surrogate progress associated with practicing a task. The task pool and embeddings remain fixed. To emphasize surrogate progress in dimensions with larger estimated proficiency gaps, we define the gap-weight vector and the corresponding scalar reward as

$$\widehat{r}_t = \widehat{\mathbf{g}}_t^\top \widehat{\boldsymbol{\delta}}_t, \quad \text{where } \widehat{\mathbf{g}}_t = \mathrm{softmax}\left(\frac{\mathbf{1} - \widehat{\mathbf{s}}_t}{\gamma_g}\right), \tag{1}$$

where $\mathbf{1} \in \mathbb{R}^K$ is the all-ones vector, $\gamma_g > 0$ is a temperature parameter, and $\mathrm{softmax}$ exponentiates each coordinate and normalizes the vector to sum to one. Among any two dimensions, the one with lower estimated proficiency receives a larger weight. Thus, equal surrogate progress receives greater weight on a dimension with a larger estimated gap. A smaller γ_g concentrates the weights more strongly on the largest gaps, whereas a larger value distributes them more evenly. Since the common unit target cancels in the normalization, the weights depend on relative proficiency differences. The gap vector is computed before action selection. Its entries are nonnegative and sum to one, therefore $\|\widehat{\mathbf{g}}_t\|_2 \le \|\widehat{\mathbf{g}}_t\|_1 = 1$. Also, $\widehat{\boldsymbol{\delta}}_t \in [-1, 1]^K$ implies $\widehat{r}_t \in [-1, 1]$, independently of K.

We consider deterministic policies based on the current maintained state, comprising the profile estimator and online task-selection statistics described in Section 3. Past observations enter decisions through these recursively updated quantities. Let Π denote the policy class and write $\pi = (\pi_t)_{t=1}^T$, where π_t is the round-specific decision map with its current internal state left implicit. Its explicit inputs are the current estimated profile and the fixed candidate-task embeddings. The objective is to maximize the expected cumulative reward over a bandit horizon of T rounds,

$$\sup_{\pi \in \Pi} \mathbb{E}_\pi \left[\sum_{t=1}^T \widehat{r}_t \right], \qquad \tau_t = \pi_t(\widehat{\mathbf{s}}_t, \{\mathbf{e}_\tau\}_{\tau \in \mathcal{T}}).$$

The expectation is over the environmental and estimation randomness along the trajectory induced by π. In the selection rule below, $\widetilde{\boldsymbol{\delta}}_t(\tau)$ denotes the ridge prediction of surrogate progress for candidate task τ, and $\widehat{\mathbf{g}}_t^\top \widetilde{\boldsymbol{\delta}}_t(\tau)$ is the corresponding predicted scalar reward.

3 METHOD

SPG-Select begins with $N \in \mathbb{N}_+$ warmup observations indexed by $i = 1, \ldots, N$, then resets the round index to $t = 1$ for the bandit phase. Warmup samples tasks uniformly within balanced randomized passes. A state-space MIRT model estimates the latent trajectory and item parameters. We train a ridge regression to predict MIRT discrimination vector and difficulty from task embeddings for every task in the task pool used for evolution, including tasks not sampled during warmup. Differences between consecutive approximate profile means smoothed with all warmup data are used to train a multilayer perceptron (MLP) that maps task embeddings and profiles to features for predicting surrogate progress. Online profile estimates instead use predictive means conditioned only on pre-decision observations. During the bandit phase, SPG-Select-R computes the gap weights $\widehat{\mathbf{g}}_t$ from $\widehat{\mathbf{s}}_t$ and scores every task using the fixed MLP encoder and ridge predictions with confidence widths. Each outcome updates the profile and surrogate-progress model, while the trajectory updates reusable backend knowledge. Ridge statistics accumulate within an epoch and reset at the next boundary. In the reported comparisons of Table 1, SPG-Select-R has higher held-out terminal success than SPG-Select-W in four of six backend-benchmark settings, ties in one, and trails in one. We therefore detail SPG-Select-R here and analyze it theoretically.

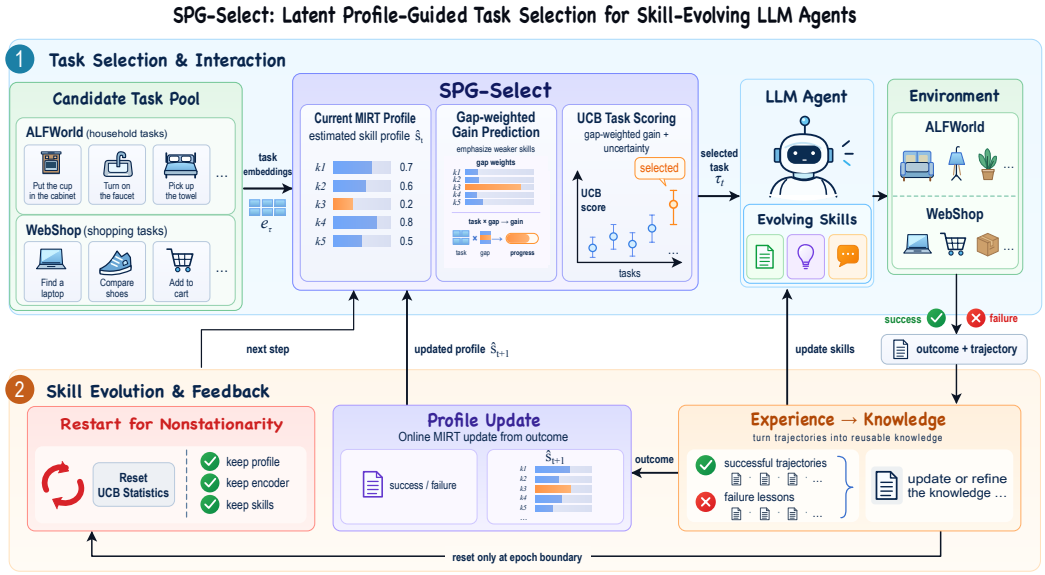


Figure 1: SPG-Select framework for latent profile-guided task selection and skill evolution.

MIRT state-space model. Each interaction yields a binary outcome, although a task may involve several skills. MIRT uses variation in task loadings across tasks to infer a common K-dimensional latent state from outcomes (Reckase, 2006). We call the predictive mean of this state $\widehat{\mathbf{s}}_t$ the profile, whose coordinates are illustrated schematically in Figure 1. The profile is therefore a model-dependent estimate of latent proficiency rather than an observed decomposition of the agent's skills. MIRT associates task τ with an unobserved discrimination vector $\mathbf{a}_\tau^\star \in \mathbb{R}^K$ and difficulty $d_\tau^\star \in \mathbb{R}$. Given the true latent state $\mathbf{s}_t^\star$, the modeled conditional success probability for task τ is

$$p_\tau^\star(\mathbf{s}_t^\star) = \mathrm{sigmoid}\big(\mathbf{a}_\tau^{\star\top}\mathbf{s}_t^\star - d_\tau^\star\big)\,,$$

where $\mathrm{sigmoid}(x) = 1/(1+e^{-x})$. This Bernoulli likelihood separates task difficulty from state-dependent proficiency. For a generic state $\mathbf{s} \in [0,1]^K$ and item parameters $(\mathbf{a}_\tau, d_\tau) \in \mathbb{R}^K \times \mathbb{R}$, let $p_\tau(\mathbf{s}) = \mathrm{sigmoid}(\mathbf{a}_\tau^\top \mathbf{s} - d_\tau)$, and denote the fitted item parameters by $(\widehat{\mathbf{a}}_\tau, \widehat{d}_\tau)$. During warmup, we fit the item parameters and latent trajectory with expectation-maximization (EM) (Dempster et al., 1977). The E-step estimates the latent-state posterior under the current item parameters. The idealized M-step maximizes a regularized expected complete-data log density over the constrained item parameters. For latent states $\mathbf{s}_i$, we use $p(\mathbf{s}_{i+1} \mid \mathbf{s}_i) = \mathcal{N}_{[0,1]^K}(\mathbf{s}_i, \mathbf{\Sigma}_s)$, where $\mathbf{\Sigma}_s \in \mathbb{R}^{K\times K}$ is fixed and positive definite, and $\mathcal{N}_{[0,1]^K}$ denotes a Gaussian distribution truncated to $[0,1]^K$. After warmup, item parameters are fixed and the MIRT belief is filtered online. Let $b_t(\mathbf{s})$ be the predictive density before round t, and let $\mathbf{s}' \in [0,1]^K$ be the next-state variable. Since the likelihood of y_t is conditioned on $\mathbf{s}$, the update first conditions on y_t and then propagates the state,

$$b_t^+(\mathbf{s}) \propto p(y_t \mid \mathbf{s}, \widehat{\mathbf{a}}_{\tau_t}, \widehat{d}_{\tau_t}) b_t(\mathbf{s}), \qquad b_{t+1}(\mathbf{s}') = \int_{[0,1]^K} p(\mathbf{s}' \mid \mathbf{s}) b_t^+(\mathbf{s})\, d\mathbf{s}, \qquad \widehat{\mathbf{s}}_{t+1} = \mathbb{E}_{b_{t+1}}[\mathbf{s}'].$$

Here, b_t^+ is the posterior for the current latent state after observing y_t, and b_{t+1} predicts the next latent state. Its mean $\widehat{\mathbf{s}}_{t+1}$ is the next profile estimate, and consecutive profiles define surrogate progress. The vector-valued profile makes the dimension-wise gap vector in equation 1 well defined. Appendix C.1 details the EM objective, supplementary model properties, and the identification and numerical choices needed to specify the estimator. Algorithm 2 in Appendix B gives the complete procedure. Notably, the profile interpretation is conditional on the fitted state-space model.

Feature construction. To predict surrogate progress, defined as the temporal difference between consecutive profile estimates, we train a two-layer MLP $\widehat{\Phi}$, where cat denotes vector concatenation. The learned encoder maps $\mathrm{cat}(\mathbf{e}_\tau, \widehat{\mathbf{s}})$ to a unit-ball feature $\boldsymbol{\phi} \in \mathbb{R}^{d_f}$ via a hidden vector $\mathbf{h} \in \mathbb{R}^{d_h}$,

where $d_h, d_f \in \mathbb{N}_+$. The fitted encoder parameters are $\widehat{\mathbf{U}}_1 \in \mathbb{R}^{d_h \times (d_e+K)}$, $\widehat{\mathbf{c}}_1 \in \mathbb{R}^{d_h}$, $\widehat{\mathbf{U}}_2 \in \mathbb{R}^{d_f \times d_h}$, and $\widehat{\mathbf{c}}_2 \in \mathbb{R}^{d_f}$. The maximum in the hidden-layer definition is applied coordinatewise:

$$\mathbf{h} = \max\Big(\widehat{\mathbf{U}}_1 \operatorname{cat}(\mathbf{e}_\tau, \widehat{\mathbf{s}}) + \widehat{\mathbf{c}}_1, 0\Big), \qquad \widetilde{\boldsymbol{\phi}} = \widehat{\mathbf{U}}_2 \mathbf{h} + \widehat{\mathbf{c}}_2, \qquad \boldsymbol{\phi} = \frac{\widetilde{\boldsymbol{\phi}}}{\max\{1, \|\widetilde{\boldsymbol{\phi}}\|_2\}}.$$

For each warmup index i, let $\widehat{\mathbf{s}}_i$ denote the approximate smoothed profile mean defined in Appendix C.1, and define the pretraining label as $\widehat{\boldsymbol{\delta}}_i = \widehat{\mathbf{s}}_{i+1} - \widehat{\mathbf{s}}_i$. These labels are constructed solely from warmup data, whereas online feedback is computed from temporal differences between consecutive predictive means. A linear pretraining head $\widehat{\mathbf{U}}_3 \in \mathbb{R}^{d_f \times K}$ maps $\boldsymbol{\phi}$ to $\widehat{\mathbf{U}}_3^\top \boldsymbol{\phi}$ and is trained to predict these labels. The normalization is part of $\widehat{\Phi}$, so $\|\widehat{\Phi}(\cdot)\|_2 \le 1$ by construction. The encoder is trained on $\{(\operatorname{cat}(\mathbf{e}_{\tau_i}, \widehat{\mathbf{s}}_i), \widehat{\boldsymbol{\delta}}_i)\}_{i=1}^N$. After pretraining, $\widehat{\mathbf{U}}_3$ is discarded and $\widehat{\Phi}$ is fixed.

Gap-weighted UCB selection. Periodic restarting is a standard strategy for handling nonstationarity in bandit problems, since it limits the influence of observations generated under earlier parameter regimes (Zhao et al., 2020). Our SPG-Select-R policy extends periodically restarted UCB selection to vector-valued surrogate progress and resets its neural linear ridge statistics at each epoch boundary. Within each epoch, gap-weighted UCB directs practice toward tasks with high predicted surrogate progress in dimensions with larger current profile gaps, while exploring tasks whose contributions remain uncertain. Fix a restart period $H \in \{1, \dots, T\}$. Let $t_0 = 1 + H\lfloor (t-1)/H \rfloor$ be the first round of the epoch containing t, and let $\mathcal{E}_t = \{t_0, \dots, t-1\}$. For each $i \in \mathcal{E}_t$, define $\boldsymbol{\phi}_i = \widehat{\Phi}(\operatorname{cat}(\mathbf{e}_{\tau_i}, \widehat{\mathbf{s}}_i))$. With ridge regularization parameter $\lambda > 0$ and $\mathbf{I}_{d_f}$ denoting the $d_f \times d_f$ identity matrix, the ridge estimate $\widehat{\mathbf{W}}_t \in \mathbb{R}^{d_f \times K}$, design matrix $\mathbf{A}_t \in \mathbb{R}^{d_f \times d_f}$, and response matrix $\mathbf{B}_t \in \mathbb{R}^{d_f \times K}$ are

$$\widehat{\mathbf{W}}_t = \mathbf{A}_t^{-1} \mathbf{B}_t, \quad \mathbf{A}_t = \lambda \mathbf{I}_{d_f} + \sum_{i \in \mathcal{E}_t} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top, \quad \mathbf{B}_t = \sum_{i \in \mathcal{E}_t} \boldsymbol{\phi}_i \widehat{\boldsymbol{\delta}}_i^\top.$$

Since $\lambda > 0$, $\mathbf{A}_t \succeq \lambda \mathbf{I}_{d_f} \succ 0$, where $\succeq$ denotes the positive-semidefinite order and $\succ 0$ denotes positive definiteness. Thus, $\mathbf{A}_t^{-1}$ is well defined. For a candidate task τ, let $\boldsymbol{\phi}_{t,\tau} = \widehat{\Phi}(\operatorname{cat}(\mathbf{e}_\tau, \widehat{\mathbf{s}}_t))$. Unlike conventional linear bandits that model directly observed scalar rewards (Abbasi-Yadkori et al., 2011), our binary observation is converted through the MIRT update into an estimated K-dimensional profile temporal difference. A task may affect several dimensions whose decision value depends on the current profile. SPG-Select therefore first predicts the vector-valued surrogate progress $\widetilde{\boldsymbol{\delta}}_t(\tau) = \widehat{\mathbf{W}}_t^\top \boldsymbol{\phi}_{t,\tau}$ and then evaluates it using the current gap vector $\widehat{\mathbf{g}}_t$. Fitting a scalar reward would combine the dimensions using the weights present when each label was formed. Vector prediction instead retains the predicted temporal difference in every profile dimension, allowing the same fitted surrogate-progress model to be evaluated using the current weights as the profile evolves. Write $w_t(\tau) = \|\boldsymbol{\phi}_{t,\tau}\|_{\mathbf{A}_t^{-1}} = \sqrt{\boldsymbol{\phi}_{t,\tau}^\top \mathbf{A}_t^{-1} \boldsymbol{\phi}_{t,\tau}}$ for the confidence width. The UCB score is

$$\mathcal{U}_t(\tau) = \widehat{\mathbf{g}}_t^\top \widetilde{\boldsymbol{\delta}}_t(\tau) + \beta_t \|\widehat{\mathbf{g}}_t\|_2 w_t(\tau).$$

The policy selects a maximizer of $\mathcal{U}_t(\tau)$, with ties broken deterministically. Its first term, labeled "gain" in Figure 1, is the predicted gap-weighted surrogate progress $\widehat{\mathbf{g}}_t^\top \widetilde{\boldsymbol{\delta}}_t(\tau)$. Its second term is a confidence bonus for this scalarized prediction, where $\beta_t > 0$ controls exploration. The theoretical guarantee uses the calibrated radius in equation 2, whereas the reported experiments use the fixed coefficient specified in Table 6. Given the fitted components and prescribed control parameters, the predictive density b_t and ridge matrices $\mathbf{A}_t, \mathbf{B}_t$ constitute the state used by the selector at round t. They determine the current profile estimate, surrogate-progress predictions, and confidence widths, and are updated recursively after each observation, with the ridge statistics reset at epoch boundaries.

Algorithm overview. Algorithms 1 and 2 in Appendix B give the complete procedures for SPG-Select-R and the profile update. At a high level, warmup calibrates the MIRT estimator and trains the task-profile representation that is fixed thereafter. Each online round converts the pre-decision profile estimate into gap weights, selects the task with the largest surrogate-progress UCB score, and uses the observed outcome to compute the next profile estimate and its temporal difference for the current epoch's ridge update. At an epoch boundary, only the ridge/UCB statistics are reset. The MIRT estimates, frozen encoder, and backend state are retained. Thus, restarting limits the influence of stale surrogate-progress observations while preserving the accumulated skill-evolution state.

4 EXPERIMENTS

Our experiments examine whether profile-guided task selection improves held-out performance across skill-evolution mechanisms, how SPG-Select-R compares with SPG-Select-W, and how UCB exploration and profile-gap weighting affect performance. We study ALFWorld and WebShop with SkillRL, ExpeL, and Shared Reflexion.

4.1 EXPERIMENTAL SETUP

Benchmarks. ALFWorld (Shridhar et al., 2020) is a text-based household environment in which agents complete tasks involving object manipulation and compositional action sequences. We evaluate on 134 tasks from its unseen split, comprising 24 Pick, 18 Look, 31 Clean, 23 Heat, 21 Cool, and 17 Pick2 tasks. This evaluation tests whether acquired knowledge transfers to unseen household situations. WebShop (Yao et al., 2022a) is a simulated shopping environment that requires agents to search for products, inspect their attributes, and purchase an item satisfying a natural-language instruction. We use its 1,000-task development pool for skill evolution and its 500-task test pool for evaluation. Together, the benchmarks cover household planning and goal-directed web interaction. Additional implementation details, computational resources, and backend-specific hyperparameters are reported in Appendix C.2.

Practice and evaluation. During practice, the task-selection policy selects a task and the agent makes one attempt using its current backend state. For each backend-benchmark comparison, every policy first receives the same N warmup interactions. The backend state produced by this common warmup initializes the ensuing T-round policy comparison. SPG-Select uses the observed outcome to compute the surrogate-progress feedback defined in Section 3, namely the temporal difference between consecutive profile estimates. This feedback is computed from the outcome before the trajectory is used for the subsequent backend update. Actor-model parameters remain fixed, so adaptation takes place through accumulated skills, insights, or reflections. After practice, we freeze the resulting backend state and evaluate each held-out task with a single attempt. No updates or test-time reflection retries are performed during evaluation. This protocol measures the transfer of acquired knowledge to held-out tasks.

Metrics. We report the terminal task success rate as a percentage of the evaluation pool. ALFWorld success is determined by the environment's task-completion signal. WebShop success requires a terminal reward of 1.0, and partial rewards do not count as successes. For WebShop, we also report Score, the mean terminal reward expressed as a percentage, which preserves partial credit. We additionally report ALFWorld results by task family to examine variation across task types. All reported gains are percentage-point differences computed from unrounded rates.

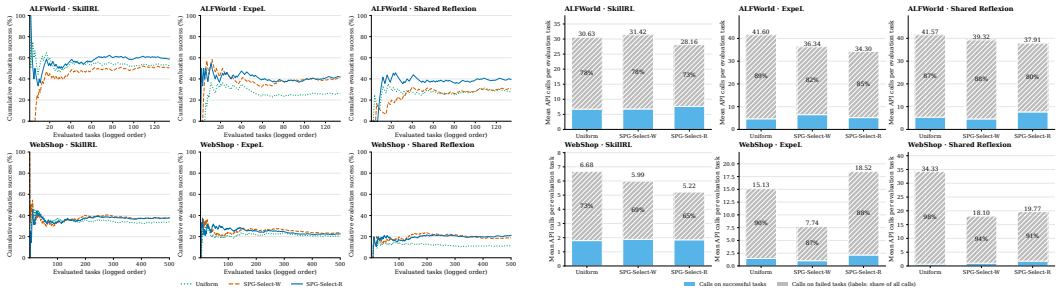
Skill-evolution backends. We integrate the skill-evolution component of three backends. SkillRL contributes its hierarchical SkillBank and failure-driven skill acquisition (Xia et al., 2026). ExpeL contributes experience-derived insights and demonstration retrieval (Zhao et al., 2024). Shared Reflexion is our cross-task memory adaptation of the original Reflexion method (Shinn et al., 2023). It replaces task-local episodic memory with a cross-task memory. Representative persisted knowledge and held-out execution traces appear in Appendix C.4.

Task-selection policies. In the tables, Uniform denotes uniform task selection without a bandit model. It samples without replacement from the full task pool within each pass and draws a new random ordering for the next pass, independently of observed outcomes or backend state. The sliding-window variant SPG-Select-W is described in Appendix B.1. Comparing SPG-Select-R with SPG-Select-W evaluates two approaches to nonstationary adaptive task selection.

Implementation fidelity. Within each backend-benchmark comparison, the task-selection policies share the practice and held-out task pools, backend update mechanism, initial skill pool, and random seed. The integrated implementation also supports backend-state features, task-eligibility rules, and backend-specific stopping conditions, as described in Appendix C.2.2.

Table 1: Held-out performance (%).

Backend	Setting	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
SkillRL	Uniform	37.5	66.7	**48.4**	65.2	**57.1**	41.2	52.2	59.4	33.8
	SPG-Select-W	41.7	**77.8**	35.5	65.2	42.9	**52.9**	50.7 (-1.5)	62.6 (+3.2)	**37.8** (+4.0)
	SPG-Select-R	**66.7**	**77.8**	45.2	**69.6**	52.4	47.1	**59.0** (+6.7)	**64.9** (+5.5)	**37.8** (+4.0)
ExpeL	Uniform	8.3	44.4	41.9	30.4	23.8	0.0	26.1	51.8	21.0
	SPG-Select-W	33.3	55.6	35.5	**47.8**	**61.9**	**5.9**	40.3 (+14.2)	**56.5** (+4.7)	**24.2** (+3.2)
	SPG-Select-R	**37.5**	**66.7**	**45.2**	39.1	57.1	0.0	**41.8** (+15.7)	46.9 (-4.9)	22.6 (+1.6)
Shared Reflexion	Uniform	16.7	55.6	38.7	17.4	33.3	0.0	27.6	23.7	11.2
	SPG-Select-W	**29.2**	44.4	29.0	**43.5**	33.3	0.0	30.6 (+3.0)	44.9 (+21.2)	18.8 (+7.6)
	SPG-Select-R	25.0	**61.1**	**51.6**	39.1	**42.9**	**11.8**	**39.6** (+11.9)	**46.0** (+22.3)	**21.2** (+10.0)



(a) Cumulative held-out success over a common evaluation order.

(b) Mean application programming interface (API) calls per held-out task, decomposed by outcome.

Figure 2: Held-out evaluation behavior with frozen backend memories.

4.2 Results

Table 1 compares the task-selection policies across backends and benchmarks, with ALFWorld results reported by task family. Each skill-evolution run requires tens of hours, so the computational budget limited evaluation to one seed per backend-benchmark comparison. In both tables, 'All' denotes the task-count-weighted ALFWorld success rate. For ALFWorld All and the two WebShop metrics in Table 1, parenthetical values give percentage-point changes from Uniform, computed from unrounded values for the same backend and metric. Bold and underlined entries mark the best and second-best values within each backend and metric, with ties receiving the same formatting.

In the reported runs, SPG-Select-R exceeds Uniform in all six backend-benchmark combinations, with gains of 6.7 to 15.7 percentage points on ALFWorld and 1.6 to 10.0 percentage points on WebShop success. SPG-Select-W exceeds Uniform in five of the six settings. For SkillRL on ALFWorld, its success rate is 1.5 percentage points lower. SPG-Select-R achieves higher success than SPG-Select-W for all three backends on ALFWorld. On WebShop, SPG-Select-R and SPG-Select-W tie in terminal success for SkillRL, SPG-Select-W performs better for ExpeL, and SPG-Select-R performs better for Shared Reflexion. WebShop Score shows a related but not identical pattern. SPG-Select-R achieves the highest Score for SkillRL and Shared Reflexion, while SPG-Select-W achieves the highest Score for ExpeL. Task-family rankings vary within ALFWorld. Relative to SPG-Select-W, SPG-Select-R has its largest gain on Pick for SkillRL and on Clean for Shared Reflexion, whereas SPG-Select-W achieves higher success on Heat for ExpeL and Shared Reflexion.

Figure 2a reports cumulative success over a common logged evaluation order. Since backend states are frozen, changes across prefixes reflect the task prefixes rather than test-time updates. Figure 2b decomposes mean API calls into contributions from success and failures, using the full evaluation-pool size as the denominator for both components. Failed tasks account for most calls, especially for ExpeL and Shared Reflexion. These costs describe the frozen agents and do not measure practice-time sample efficiency. Appendix C.2.3 reports total API calls and comparisons with Uniform.

Table 2: ALFWorld ablations (%). All deltas vs. SPG-Select-R.

Backend	Setting	Pick	Look	Clean	Heat	Cool	Pick2	All
SkillRL	Uniform	37.5	66.7	48.4	65.2	57.1	41.2	52.2 (-6.7)
	w/o UCB	50.0	66.7	38.7	65.2	**61.9**	35.3	52.2 (-6.7)
	w/o gap	50.0	**77.8**	**54.8**	65.2	57.1	35.3	56.7 (-2.2)
	$K = 1$	50.0	72.2	**54.8**	**73.9**	52.4	35.3	56.7 (-2.2)
	Full history	58.3	61.1	41.9	69.6	47.6	41.2	53.0 (-6.0)
	SPG-Select-R	**66.7**	**77.8**	45.2	69.6	52.4	**47.1**	**59.0**
ExpeL	Uniform	8.3	44.4	41.9	30.4	23.8	0.0	26.1 (-15.7)
	w/o UCB	33.3	44.4	35.5	52.2	47.6	0.0	36.6 (-5.2)
	w/o gap	33.3	38.9	35.5	**56.5**	**57.1**	**17.6**	40.3 (-1.5)
	$K = 1$	29.2	**83.3**	29.0	43.5	19.0	0.0	33.6 (-8.2)
	Full history	**37.5**	55.6	**45.2**	39.1	**57.1**	5.9	41.0 (-0.7)
	SPG-Select-R	**37.5**	66.7	**45.2**	39.1	**57.1**	0.0	**41.8**
Shared Reflexion	Uniform	16.7	55.6	38.7	17.4	33.3	0.0	27.6 (-11.9)
	w/o UCB	33.3	**61.1**	29.0	**39.1**	42.9	5.9	35.1 (-4.5)
	w/o gap	8.3	55.6	19.4	17.4	28.6	0.0	20.9 (-18.7)
	$K = 1$	25.0	50.0	25.8	34.8	42.9	0.0	29.9 (-9.7)
	Full history	**37.5**	50.0	38.7	13.0	**47.6**	0.0	32.1 (-7.5)
	SPG-Select-R	25.0	**61.1**	**51.6**	**39.1**	42.9	**11.8**	**39.6**

Appendix C.2.1 shows two sets of experiments that together support the use of gap-weighted reward as a feedback signal for skill evolution.

4.3 Ablation Study

Table 2 combines ALFWorld component ablations with profile-dimension and history ablations. Every ablation uses exactly the same practice and held-out task pools as its corresponding main experiment, so the comparisons are not affected by differences in task-pool composition. Here, w/o UCB means without the UCB bonus, and w/o gap means without gap weighting. w/o UCB sets $\beta_t = 0$, removing the confidence bonus while retaining the gap-weighted prediction. w/o gap sets $\widehat{\mathbf{g}}_t = \mathbf{1}/K$ in the selection score while retaining $\beta_t = 0.1$. It therefore weights skill dimensions equally while preserving the confidence bonus and the estimated profile in the context. Relative to SPG-Select-R, these ablations isolate the confidence bonus and gap weighting, respectively. The direct comparison between w/o UCB and w/o gap differs in both components. The $K = 1$ setting uses a scalar latent profile with periodic restarting, whereas Full history uses the $K = 4$ profile while retaining all ridge observations. Parentheses in the All column indicate percentage-point changes from SPG-Select-R, computed from unrounded rates.

On ALFWorld, SPG-Select-R improves aggregate success over w/o UCB by 6.7, 5.2, and 4.5 percentage points for SkillRL, ExpeL, and Shared Reflexion. Its gains over w/o gap are 2.2, 1.5, and 18.7 points, respectively. Relative to the scalar-profile $K = 1$ setting, the gains are 2.2, 8.2, and 9.7 points. Relative to Full history, they are 6.0, 0.7, and 7.5 points. The w/o UCB and w/o gap results are consistent with benefits from uncertainty-driven exploration and profile-gap weighting, respectively. The lower aggregate success under $K = 1$ and Full history is consistent with benefits from a multidimensional profile and periodically resetting the ridge statistics. The task-family results vary, but SPG-Select-R has the highest aggregate success for each backend. We also evaluate the ablations on WebShop, where SPG-Select-R achieves higher success than the ablations. To keep the main text concise, detailed results and metric-level analysis appear in Appendix C.2 (Table 7).

5 THEORETICAL ANALYSIS

We analyze vector-valued SPG-Select-R under path variation, following the analysis of Zhao et al. (2020). All statements condition on the warmup outputs, with the task pool $\mathcal{T}$, encoder $\widehat{\Phi}$, and MIRT item parameters fixed. The model and noise assumptions in Appendix D, which also provides the proofs, must hold under this conditioning. The guarantee concerns surrogate feedback under exact linear realizability, which is a separate modeling assumption beyond encoder pretraining. Let $\mathcal{F}_{t-1}$ denote the sigma-field generated by the warmup outputs and all tasks, outcomes, trajectories, backend updates, and estimator randomness available before round t. For each candidate τ, let $\widehat{\boldsymbol{\delta}}_t(\tau)$ be its potential surrogate progress, i.e., the potential temporal difference of the profile estimate. Only $\widehat{\boldsymbol{\delta}}_t(\tau_t)$ is observed. Define $\boldsymbol{\phi}_{t,\tau} = \widehat{\Phi}(\mathrm{cat}(\mathbf{e}_\tau, \widehat{\mathbf{s}}_t))$ and the conditional mean surrogate progress $\boldsymbol{\mu}_t(\tau) = \mathbb{E}[\widehat{\boldsymbol{\delta}}_t(\tau) \mid \mathcal{F}_{t-1}]$. The conditional mean scalar reward is $q_t(\tau) = \widehat{\mathbf{g}}_t^\top \boldsymbol{\mu}_t(\tau)$.

Definition 5.1. Let $\tau_t^* \in \arg\max_{\tau \in \mathcal{T}} q_t(\tau)$ be a dynamic oracle action. The dynamic pseudo-regret of a deterministic policy over T bandit rounds is defined as

$$R_T = \sum_{t=1}^{T} \big[q_t(\tau_t^*) - q_t(\tau_t)\big].$$

Both the oracle action and the selected action are evaluated along the learner's realized history at the same current profile and backend state. Thus, R_T measures cumulative one-step allocation quality along the realized history. Under Assumptions D.1 and D.2, the conditional mean model is $\boldsymbol{\mu}_t(\tau) = \mathbf{W}_t^{\star\top} \boldsymbol{\phi}_{t,\tau}$, with $\mathbf{W}_t^\star \in \mathbb{R}^{d_f \times K}$. Definition D.3 gives its path length as $P_T = \sum_{t=1}^{T-1} \|\mathbf{W}_{t+1}^\star - \mathbf{W}_t^\star\|_F$. Restarting restricts the ridge statistics to the current epoch. For fixed $H \in \{1, \ldots, T\}$ and $\rho \in (0, 1)$, the confidence radius β_t is defined in equation 2, and $\beta = \max_{1 \le t \le T} \beta_t$.

Theorem 5.2. *Under Assumptions D.1 and D.2, with path length P_T as in Definition D.3, fix $H \in \{1, \ldots, T\}$ before the bandit phase, let $\rho \in (0, 1)$, and define β_t and β by equation 2. Then, conditional on the warmup outputs, with probability at least $1 - \rho$, the deterministic UCB policy specified in Algorithm 1 of Appendix B satisfies*

$$R_T \le 2\beta \left\lceil \frac{T}{H} \right\rceil \sqrt{\frac{(\lambda+1) d_f H}{\lambda} \log\left(1 + \frac{H}{\lambda d_f}\right)} + 2\sqrt{\frac{d_f}{\lambda}} H^{3/2} P_T.$$

Lemma D.6 bounds prediction error by statistical uncertainty and within-epoch drift, simultaneously for all candidate tasks and rounds on one event of probability at least $1 - \rho$. This uniformity permits adaptive task selection. The selection score uses the statistical bonus, while the proof accounts separately for drift. The theorem's first term is the estimation cost accumulated across restarted epochs, and its second is the bias from within-epoch parameter drift. The bound holds for any fixed restart period and uses the realized, potentially random path length P_T. For oracle tuning, Corollary D.8 instead assumes a deterministic budget $V_T \ge 0$, specified before the bandit phase, such that $P_T \le V_T$ almost surely under the resulting policy. Let β^{u} be the horizon-uniform radius in equation 3. For $V_T > 0$, define $H^\circ = (\beta^{\mathrm{u}} T \sqrt{(\lambda+1)\log(1 + T/(\lambda d_f))}/V_T)^{1/2}$, $H^* = \min\{T, \max\{1, \lceil H^\circ \rceil\}\}$. Run the policy with $H = H^*$ and the radii in equation 2. If $1 \le H^\circ \le T$, the corollary gives, conditional on the warmup outputs, with probability at least $1 - \rho$, $R_T = \mathcal{O}(d_f^{1/2}(\lambda+1)^{3/8} \log^{3/8}(1 + T/(\lambda d_f))(\beta^{\mathrm{u}} T)^{3/4} V_T^{1/4}/\lambda^{1/2})$. Let S and σ denote the parameter-norm bound and noise scale in Assumption D.2. When $K, \lambda, \lambda^{-1}, S, \sigma, \rho^{-1}$ are fixed independently of d_f and T, this becomes $R_T = \widetilde{\mathcal{O}}(d_f^{7/8} T^{3/4} V_T^{1/4})$. For $V_T = 0$, choosing $H = T$ gives $R_T = \widetilde{\mathcal{O}}(d_f \sqrt{T})$ in the same regime. The oracle choice uses an a priori variation budget, and substituting the future realized P_T would not define a nonanticipating policy. The detailed discussion and proof appear in Appendix D.

6 CONCLUSION

We present SPG-Select for finite-budget task selection in skill-evolving LLM agents. It combines MIRT profile estimation, gap-weighted surrogate-progress prediction, and uncertainty-aware selection. In the reported single-seed comparisons, SPG-Select-R improves held-out success over Uniform in all six backend-benchmark combinations. Under the stated surrogate model, it admits a high-probability dynamic pseudo-regret bound for surrogate task allocation.

Artificial Intelligence Use Statement

We used generative artificial intelligence (AI) tools to assist with language editing, LaTeX formatting, code development and debugging, and figure preparation. We also used AI-generated feedback to identify potential weaknesses in our research ideas, theoretical derivations, and experiments. The authors have reviewed and verified all AI-assisted text, code, mathematical arguments, and figures, and retain full responsibility for the final content of this work.

Ethics Statement

This work does not involve human participants or the release of personal data. Its task-selection mechanism may affect how LLM agents allocate interaction budgets, so deployment decisions should consider task pools, downstream capabilities, and potential misuse in the target environment. The method should be assessed for safety and fairness before deployment in settings where agent actions can affect people or resources.

Reproducibility Statement

The problem formulation, MIRT update, feature construction, and decision rule are specified in Sections 2 and 3, while the theoretical setup and guarantees are given in Section 5. Appendix C.2 reports computational resources and backend-specific hyperparameters, Appendix C.2.2 documents backend integration details, and Appendix C.4 provides representative persisted knowledge and held-out execution traces. Algorithms 1 and 2 in Appendix B give the complete procedures, while Appendix D provides the detailed proofs of the theoretical claims. The theoretical derivations are self-contained.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

Yuxuan Cai, Yipeng Hao, Jie Zhou, Hang Yan, Zhikai Lei, Rui Zhen, Zhenhua Han, Yutao Yang, Junsong Li, Qianjun Pan, et al. Building self-evolving agents via experience-driven lifelong learning: A framework and benchmark. *arXiv preprint arXiv:2508.19005*, 2025.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Learning to optimize under non-stationarity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1079–1087. PMLR, 2019.

Sunmook Choi, Yahya Sattar, Yassir Jedra, Maryam Fazel, and Sarah Dean. Explore-then-commit for nonstationary linear bandits with latent dynamics. *arXiv preprint arXiv:2510.16208*, 2025.

Giulia Clerici, Pierre Laforgue, and Nicolò Cesa-Bianchi. Linear bandits with memory. *Transactions on Machine Learning Research*, 2024.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Susanne Frick, Amer Krivosija, and Alexander Munteanu. Scalable learning of item response theory models. In *International Conference on Artificial Intelligence and Statistics*, pp. 1234–1242. PMLR, 2024.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

Mark D Reckase. 18 multidimensional item response theory. *Handbook of statistics*, 26:607–642, 2006.

Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. *Advances in Neural Information Processing Systems*, 32, 2019.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

James Sharpnack, Kevin Hao, Phoebe Mulcaire, Klinton Bicknell, Geoff LaFlair, Kevin Yancey, and Alina A von Davier. Banditcat and autoirt: Machine learning approaches to computerized adaptive testing and item calibration. *arXiv preprint arXiv:2410.21033*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*, 2025.

Sang Truong, Yuheng Tu, Percy Liang, Bo Li, and Sanmi Koyejo. Reliable and efficient amortized model-based evaluation. *arXiv preprint arXiv:2503.13335*, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Zhiyong Wang, Jize Xie, Yi Chen, John Lui, and Dongruo Zhou. Variance-dependent regret bounds for non-stationary linear bandits. *arXiv preprint arXiv:2403.10732*, 2024.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Zhihan Xiong, Romain Camilleri, Maryam Fazel, Lalit Jain, and Kevin Jamieson. A/b testing and best-arm identification for linear bandits with robustness to non-stationarity. In *International Conference on Artificial Intelligence and Statistics*, pp. 1585–1593. PMLR, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. A simple approach for non-stationary linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 746–755. PMLR, 2020.

Appendix

CONTENTS

A Related Work **14**

B Algorithms **14**

B.1 Sliding-Window Comparator 16

C Implementation and Experimental Details **16**

C.1 MIRT State-Space Estimation 16

C.2 Experimental Details 17

C.3 Prompt Templates and a Newly Generated Skill 22

C.4 Qualitative Examples 24

D Proofs for Theoretical Analysis **26**

D.1 Assumptions 26

D.2 Lemmas for Theorem 5.2 27

D.3 Proof of Theorem 5.2 31

D.4 Oracle Choice of the Restart Period 31

A Related Work

Experience-based skill evolution. Experience-based LLM agents can improve behavior without necessarily modifying model parameters. ReAct (reasoning and acting) provides an interface that interleaves these processes (Yao et al., 2022b), Reflexion uses linguistic self-feedback (Shinn et al., 2023), ExpeL distills transferable insights (Zhao et al., 2024), and Voyager couples a generated curriculum with an executable skill library (Wang et al., 2023). Recent work further studies experience sharing, refinement, recursive evolution, and learnable memory skills (Tang et al., 2025; Wu et al., 2025; Xia et al., 2026; Zhang et al., 2026). SPG-Select complements these knowledge-acquisition mechanisms with profile-conditioned practice-task selection. Voyager also selects tasks, using an LLM-generated curriculum rather than the surrogate-progress bandit considered here.

MIRT-based ability estimation. MIRT represents ability with a latent vector and tasks with dimension-specific discrimination parameters (Reckase, 2006). Scalable item response theory (IRT) estimation (Frick et al., 2024) and IRT-based language-model evaluation (Truong et al., 2025) extend this perspective to modern machine-learning settings. BanditCAT combines IRT with a contextual bandit for computerized adaptive testing, using Fisher information as the item-selection reward (Sharpnack et al., 2024). In contrast, SPG-Select filters a profile, defined as the MIRT latent-state estimate, and uses its temporal difference as surrogate-progress feedback for practice-task selection.

Contextual and nonstationary bandits. Contextual bandits use side information and bandit feedback for sequential action choice (Abbasi-Yadkori et al., 2011). In contrast, prioritized experience replay selects stored transitions (Schaul et al., 2015). For nonstationarity, discounted UCB (Garivier & Moulines, 2008), sliding-window UCB (Cheung et al., 2019), weighted estimation (Russac et al., 2019), and restarted UCB (Zhao et al., 2020) handle parameter drift. Recent work gives variance-dependent regret bounds (Wang et al., 2024), reward dependence on recent actions (Clerici et al., 2024), latent action-dependent dynamics (Choi et al., 2025), and fixed-budget identification (Xiong et al., 2024). Our setting instead uses task-profile contexts and estimated vector-valued surrogate progress, motivating our vector-valued restarted selector SPG-Select-R for this surrogate model.

B Algorithms

In this paper, we use $t \in 1, \ldots, T$ to index bandit steps and $i \in 1, \ldots, N$ to index warmup states. With a slight abuse of notation, we write $\widehat{s}_i$ during the warmup phase and $\widehat{s}_t$ during the bandit phase. Algorithm 1 specifies the end-to-end SPG-Select workflow. Its inputs separate quantities fixed before online interaction, including the task pool, warmup budget, feature dimension, and MIRT skill dimension, from the online-control parameters H, γ_g, and λ, and the rule that computes β_t from the current design matrix or sets it to a fixed experimental value. In the warmup phase, balanced randomized task selection produces calibration outcomes. These outcomes are used to fit the MIRT state-space model and construct surrogate-progress labels, each equal to a temporal difference between consecutive profile estimates, for the normalized encoder. The item parameters and encoder are then fixed, and the algorithm initializes the predictive density, its mean, and an empty regularized ridge head.

In the bandit phase, Algorithm 1 implements the vector-valued SPG-Select-R policy described in Section 3. It first checks whether a new epoch has begun. If so, it resets only $\mathbf{A}_t$, $\mathbf{B}_t$, and $\widehat{\mathbf{W}}_t$. The MIRT item parameters, filtering state, encoder, and backend-specific skill-evolution state persist across epochs. For every task $\tau \in \mathcal{T}$, it constructs a task-profile feature, predicts vector-valued surrogate progress, and evaluates the gap-weighted UCB score. The selected maximizing task is executed. Algorithm 2 supplies the next predictive density and its mean from the observed outcome. By definition, the temporal difference between consecutive predictive means is the surrogate-progress vector, which updates the current epoch's ridge statistics. The execution trajectory is then used for the backend memory update. These quantities constitute the maintained selector state. The task pool stays fixed as the skill-evolution state and profile estimates change across rounds.

Algorithm 2 gives the filtering operation used after each selected interaction. It first reweights the pre-observation density b_t by the selected item's MIRT likelihood and normalizes the resulting posterior b_t^+. It then propagates this posterior through the state-transition model to obtain the pre-decision density b_{t+1} for the next round. The returned predictive mean $\widehat{\mathbf{s}}_{t+1}$, which is the next

Algorithm 1: SPG-Select-R

Input. Task pool $\mathcal{T}$, warmup length $N \in \mathbb{N}_+$, bandit horizon $T \in \mathbb{N}_+$, restart period $H \in \{1, \dots, T\}$, feature dimension $d_f \in \mathbb{N}_+$, gap temperature $\gamma_g > 0$, confidence-radius rule returning $\beta_t \geq 0$, regularizer $\lambda > 0$, skill dimension $K \in \mathbb{N}_+$
Phase 1. Warmup
for $i = 1$ to N **do**
  Select τ_i by balanced randomized sampling
  Execute τ_i, observe $y_i \in \{0, 1\}$, and update the backend memory
  Store (τ_i, y_i)
end for
Fit the MIRT state-space model on $\{(\tau_i, y_i)\}_{i=1}^{N}$; fit an embedding-to-item-parameter ridge regression and predict parameters for every task in the task pool used for evolution; freeze $\{\widehat{\mathbf{a}}_\tau, \widehat{d}_\tau\}_{\tau \in \mathcal{T}}$ and construct surrogate-progress temporal-difference labels
Train and freeze $\widehat{\Phi}$ on the warmup data
Initialize b_1 and the profile $\widehat{\mathbf{s}}_1$ from the fitted warmup marginal of $\mathbf{s}_{N+1}$, reset the round index, and set $\mathbf{A}_1 = \lambda \mathbf{I}_{d_f}$, $\mathbf{B}_1 = \mathbf{0}$, and $\widehat{\mathbf{W}}_1 = \mathbf{0}$
Phase 2. Bandit
for $t = 1$ to T **do**
  if $t > 1$ and $(t-1) \bmod H = 0$ **then**
    $\mathbf{A}_t \leftarrow \lambda \mathbf{I}_{d_f}$, $\mathbf{B}_t \leftarrow \mathbf{0}$, and $\widehat{\mathbf{W}}_t \leftarrow \mathbf{0}$
  end if
  Compute $\widehat{\mathbf{g}}_t = \mathrm{softmax}((\mathbf{1} - \widehat{\mathbf{s}}_t)/\gamma_g)$
  for each $\tau \in \mathcal{T}$ **do**
    $\boldsymbol{\phi}_{t,\tau} = \widehat{\Phi}(\mathrm{cat}(\mathbf{e}_\tau, \widehat{\mathbf{s}}_t))$, $\widetilde{\boldsymbol{\delta}}_t(\tau) = \widehat{\mathbf{W}}_t^\top \boldsymbol{\phi}_{t,\tau}$
    $\mathcal{U}_t(\tau) \leftarrow \widehat{\mathbf{g}}_t^\top \widetilde{\boldsymbol{\delta}}_t(\tau) + \beta_t \|\widehat{\mathbf{g}}_t\|_2 \sqrt{\boldsymbol{\phi}_{t,\tau}^\top \mathbf{A}_t^{-1} \boldsymbol{\phi}_{t,\tau}}$
  end for
  Select $\tau_t \in \arg\max_{\tau \in \mathcal{T}} \mathcal{U}_t(\tau)$, breaking ties deterministically
  Execute τ_t and observe y_t and the execution trajectory
  $(b_{t+1}, \widehat{\mathbf{s}}_{t+1}) \leftarrow \textsc{BayesUpdate}(b_t, \widehat{\mathbf{a}}_{\tau_t}, \widehat{d}_{\tau_t}, y_t)$
  $\widehat{\boldsymbol{\delta}}_t \leftarrow \widehat{\mathbf{s}}_{t+1} - \widehat{\mathbf{s}}_t$ and $\boldsymbol{\phi}_t \leftarrow \boldsymbol{\phi}_{t,\tau_t}$
  $\mathbf{A}_{t+1} \leftarrow \mathbf{A}_t + \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\top$
  $\mathbf{B}_{t+1} \leftarrow \mathbf{B}_t + \boldsymbol{\phi}_t \widehat{\boldsymbol{\delta}}_t^\top$
  $\widehat{\mathbf{W}}_{t+1} \leftarrow \mathbf{A}_{t+1}^{-1} \mathbf{B}_{t+1}$
  Update the backend memory using the execution trajectory
end for

Algorithm 2: BayesUpdate

Input. b_t, $(\widehat{\mathbf{a}}_{\tau_t}, \widehat{d}_{\tau_t})$, and y_t
$b_t^+(\mathbf{s}) \leftarrow p(y_t \mid \mathbf{s}, \widehat{\mathbf{a}}_{\tau_t}, \widehat{d}_{\tau_t}) b_t(\mathbf{s})$
Normalize b_t^+ over $[0,1]^K$
$b_{t+1}(\mathbf{s}') \leftarrow \int_{[0,1]^K} p(\mathbf{s}' \mid \mathbf{s}) b_t^+(\mathbf{s})\, d\mathbf{s}$
$\widehat{\mathbf{s}}_{t+1} \leftarrow \mathbb{E}_{b_{t+1}}[\mathbf{s}']$
Output. b_{t+1} and $\widehat{\mathbf{s}}_{t+1}$

profile estimate, is consequently based only on observations through round t. This preserves the nonanticipating ordering required by the selector. In practice, the same constrained Laplace or variational approximation used for the MIRT state-space model can implement the normalization, propagation, and predictive-mean calculation.

B.1 SLIDING-WINDOW COMPARATOR

The sliding-window comparator SPG-Select-W uses the same frozen encoder, gap weights, and UCB score as SPG-Select-R. Following sliding-window UCB (Cheung et al., 2019), it replaces the restart index set $\mathcal{E}_t$ in the ridge sums of Section 3 by $\mathcal{E}_t^{\mathrm{W}} = \{i \in \{1, \dots, t-1\} : i \geq t - H_{\mathrm{W}}\}$ for a fixed window length $H_{\mathrm{W}} \in \{1, \dots, T\}$. After the window fills, each new observation displaces the oldest one, whereas SPG-Select-R clears the ridge statistics at epoch boundaries. Both policies retain the MIRT filtering state, frozen encoder, and backend knowledge.

C IMPLEMENTATION AND EXPERIMENTAL DETAILS

C.1 MIRT STATE-SPACE ESTIMATION

This subsection gives the rationale and warmup fitting details for the state-space MIRT model in Section 3.

Why MIRT matches task selection. The selector observes a binary environmental outcome $y_t \in \{0, 1\}$ after each chosen task. This is precisely the observation type modeled by the Bernoulli response law

$$\Pr(y_t = 1 \mid \mathbf{s}, \mathbf{a}_{\tau_t}, d_{\tau_t}) = \mathrm{sigmoid}\big(\mathbf{a}_{\tau_t}^\top \mathbf{s} - d_{\tau_t}\big)\,.$$

Here, $\mathbf{s} \in [0,1]^K$ is the agent's multidimensional latent state, $\mathbf{a}_{\tau_t} \in \mathbb{R}^K$ specifies how the selected task loads on its coordinates, and $d_{\tau_t} \in \mathbb{R}$ is the task difficulty. A single failure does not by itself distinguish a difficult task from a low discrimination-weighted latent state. The model represents these two effects with separate parameters, which are fitted jointly using warmup outcomes under the identification constraints described below. A low success rate alone is therefore not treated as evidence that any particular profile coordinate is weak.

This separation is needed for allocation. A scalar competence estimate could summarize whether the agent is performing poorly overall, but it would not identify which capability currently has the larger estimated gap. MIRT instead yields $\widehat{\mathbf{s}}_t \in [0,1]^K$, so the selector can form the dimension-wise weights $\widehat{\mathbf{g}}_t = \mathrm{softmax}((\mathbf{1} - \widehat{\mathbf{s}}_t)/\gamma_g)$ defined in Section 2. Thus, the latent profile is the state that connects binary feedback to the profile-conditioned allocation rule, rather than an interchangeable scalar summary.

State dynamics and online filtering. For generic item parameters $(\mathbf{a}_\tau, d_\tau)$, write $p_\tau(\mathbf{s}) = \mathrm{sigmoid}(\mathbf{a}_\tau^\top \mathbf{s} - d_\tau)$; their fitted values are denoted by $(\widehat{\mathbf{a}}_\tau, \widehat{d}_\tau)$. Each discrimination coefficient determines how one latent-state coordinate enters a task's predicted success probability. Because $\widehat{\mathbf{a}}_\tau$ is constrained to be elementwise nonnegative, the fitted response $\mathrm{sigmoid}(\widehat{\mathbf{a}}_\tau^\top \mathbf{s} - \widehat{d}_\tau)$ is nondecreasing in every latent-state coordinate. Any nonnegative task-pool average of the fitted responses is therefore also nondecreasing coordinatewise. Anchor items or equivalent constraints fix the latent-state orientation and scale.

For latent states $\mathbf{s}_i$, we use $p(\mathbf{s}_{i+1} \mid \mathbf{s}_i) = \mathcal{N}_{[0,1]^K}(\mathbf{s}_i, \mathbf{\Sigma}_s)$, where $\mathbf{\Sigma}_s \in \mathbb{R}^{K \times K}$ is a fixed positive-definite covariance matrix and $\mathcal{N}_{[0,1]^K}$ is truncated to $[0,1]^K$. This transition regularizes the latent trajectory and adjacent profile estimates independently of the selected task. After collecting $N \in \mathbb{N}_+$ warmup outcomes $\mathcal{D}_N = \{(\tau_i, y_i)\}_{i=1}^N$, we fit the item parameters and latent trajectory with expectation-maximization (EM) (Dempster et al., 1977). After warmup, the fitted item parameters are fixed and the MIRT belief is filtered online. Let $b_t(\mathbf{s})$ be the predictive density before round t, and let $\mathbf{s}' \in [0,1]^K$ be the next-state variable. Since y_t is generated from $\mathbf{s}$, the update first conditions on y_t and then propagates the state:

$$b_t^+(\mathbf{s}) \propto p(y_t \mid \mathbf{s}, \widehat{\mathbf{a}}_{\tau_t}, \widehat{d}_{\tau_t}) b_t(\mathbf{s}), \qquad b_{t+1}(\mathbf{s}') = \int_{[0,1]^K} p(\mathbf{s}' \mid \mathbf{s}) b_t^+(\mathbf{s})\, d\mathbf{s}, \qquad \widehat{\mathbf{s}}_{t+1} = \mathbb{E}_{b_{t+1}}[\mathbf{s}'].$$

Here, b_t^+ is the posterior for the current latent state after observing y_t, and b_{t+1} predicts the next latent state. Its mean $\widehat{\mathbf{s}}_{t+1}$, the next profile estimate, is available before the next selection. Algorithm 2 gives the complete filtering procedure.

Table 3: Common computational resources and evaluation protocol.

Item	Value
Graphics processing units (GPUs)	2× NVIDIA RTX A6000
Task-embedding model	Qwen3-Embedding-0.6B; frozen
Rollouts / interaction	1
Eval. attempts / task	1
ALFWorld $H_{\max}$	50
WebShop $H_{\max}$	50

The initial-state prior is centered at the vector whose coordinates all equal 0.5. Let $\boldsymbol{\theta} = \{(\mathbf{a}_\tau, d_\tau)\}_{\tau \in \mathcal{T}}$ collect the generic item parameters. For $y \in \{0, 1\}$, the Bernoulli mass function is

$$p(y \mid \mathbf{s}, \mathbf{a}_\tau, d_\tau) = p_\tau(\mathbf{s})^y [1 - p_\tau(\mathbf{s})]^{1-y}.$$

Together with the transition model above, this gives the complete-data log density

$$\ell_c(\boldsymbol{\theta}; \mathbf{s}_{1:N+1}, \mathcal{D}_N) = \log p(\mathbf{s}_1) + \sum_{i=1}^{N} \log p(\mathbf{s}_{i+1} \mid \mathbf{s}_i) + \sum_{i=1}^{N} \left[y_i \log p_{\tau_i}(\mathbf{s}_i) + (1 - y_i) \log\big(1 - p_{\tau_i}(\mathbf{s}_i)\big) \right].$$

The idealized fitting procedure uses EM. At iteration m, the idealized E-step forms the latent-state posterior and its means

$$\zeta^{(m)}(\mathbf{s}_{1:N+1}) = p(\mathbf{s}_{1:N+1} \mid \mathcal{D}_N, \boldsymbol{\theta}^{(m)}), \qquad \widehat{\mathbf{s}}_i^{(m)} = \mathbb{E}_{\zeta^{(m)}}[\mathbf{s}_i].$$

When the maximum is attained, the corresponding exact M-step solves

$$\boldsymbol{\theta}^{(m+1)} \in \arg\max_{\boldsymbol{\theta} \in \Theta} \left\{ \mathbb{E}_{\zeta^{(m)}}[\ell_c(\boldsymbol{\theta}; \mathbf{s}_{1:N+1}, \mathcal{D}_N)] - \frac{\eta}{2} \sum_{\tau \in \mathcal{T}} \|\mathbf{a}_\tau\|_2^2 \right\},$$

where $\eta > 0$ is the item-parameter regularization coefficient and Θ imposes the nonnegativity and identification constraints. A constrained Laplace or variational smoother can approximate the non-conjugate E-step, and the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with box constraints (L-BFGS-B) (Byrd et al., 1995) can approximately solve an M-step whose constraints are represented by box bounds. With an exact E-step, an approximate M-step is a generalized-EM update if it increases the penalized expected complete-data objective. An approximate E-step requires a separate analysis of the chosen approximation to establish monotonicity. The notation $\widehat{\mathbf{s}}_i$ denotes the resulting approximate smoothed mean at warmup index i. These warmup labels may use all N warmup outcomes. Online estimates use only observations available before each decision, and the fitted marginal of $\mathbf{s}_{N+1}$ initializes the bandit phase.

After warmup fitting, a multi-output ridge regression maps embeddings to their fitted discrimination vectors and difficulties. It assigns predictions to every task in the task pool used for evolution, including tasks not sampled during warmup; these are projected to nonnegative values and held fixed during online selection. Thus, unobserved tasks do not require independently fitted MIRT item parameters.

C.2 Experimental Details

Tables 3 and 4 summarize the common experimental protocol and backend-specific configurations. WebShop component ablations appear in Table 7.

Table 4: Backend-specific hyperparameters.

Backend	Component	Value
SkillRL	Response length	$L = 512$ tokens
	Skill retrieval	top-$k = 6$
	Skill update	interval $= 5$, threshold $\theta = 0.4$, max. new skills $= 3$
ExpeL	Embedder / retrieval	Qwen3-Embedding-0.6B; task-similarity top-$k = 2$
	Decode	actor / evaluation temperature $= 0.3$; reflection / insight temperature $= 0$
	Reflection	max. retries $= 3$
	Environment	success examples $n = (8, 8)$ for (ALFWorld, WebShop)
Shared Reflexion	Episodic memory	per-task capacity $C = 3$ reflections
	Trial budget	max. trials / task $= 3$

Table 5: Language models used by the integrated backends.

Backend	Task execution and evaluation	Skill evolution
SkillRL	SkillRL-trained Qwen2.5-7B-Instruct	DeepSeek-V4-Flash
ExpeL	Qwen2.5-7B-Instruct	DeepSeek-V4-Flash
Shared Reflexion	Qwen2.5-7B-Instruct	DeepSeek-V4-Flash

Table 6: SPG-Select hyperparameters.

Hyperparameter	K	d_f	λ	β_t	γ_g	H	N	T
Value	4	16	1.0	0.1	0.1	50	56 (ALFWorld) / 400 (WebShop)	84 (ALFWorld) / 600 (WebShop)

Table 7: WebShop ablations (%). Deltas vs. SPG-Select-R.

Backend	Setting	Score	Succ.
SkillRL	Uniform	59.4 (-5.5)	33.8 (-4.0)
	w/o UCB	62.0 (-2.9)	32.8 (-5.0)
	w/o gap	<u>64.3</u> (-0.6)	<u>35.6</u> (-2.2)
	SPG-Select-R	**64.9**	**37.8**
ExpeL	Uniform	51.8 (+4.9)	<u>21.0</u> (-1.6)
	w/o UCB	<u>52.2</u> (+5.3)	20.2 (-2.4)
	w/o gap	**54.8** (+7.8)	19.4 (-3.2)
	SPG-Select-R	46.9	**22.6**
Shared Reflexion	Uniform	23.7 (-22.3)	11.2 (-10.0)
	w/o UCB	<u>45.1</u> (-0.9)	17.6 (-3.6)
	w/o gap	44.9 (-1.1)	<u>20.0</u> (-1.2)
	SPG-Select-R	**46.0**	**21.2**

Table 7 reports the WebShop component ablations. SPG-Select-R improves terminal success over w/o UCB by 5.0, 2.4, and 3.6 percentage points for SkillRL, ExpeL, and Shared Reflexion, respectively, and over w/o gap by 2.2, 3.2, and 1.2 points. Score does not follow terminal success uniformly. Relative to w/o gap, SPG-Select-R is 0.6 points higher for SkillRL, 7.8 points lower for ExpeL, and 1.1 points higher for Shared Reflexion.

In Table 3, $H_{\max}$ denotes the maximum number of environment steps. In Table 4, L denotes the response-length cap, k the retrieval count, θ the SkillRL update threshold, n the ExpeL successful-example group size, and C Reflexion episodic-memory capacity. Parenthesized pairs follow the order (ALFWorld, WebShop).

Table 5 separates the model that acts in the environment during practice and held-out evaluation from the model that updates reusable knowledge. All SkillRL experiments use the Qwen2.5-7B-Instruct checkpoint trained by the original SkillRL pipeline for task execution, whereas the ExpeL and Shared Reflexion experiments use Qwen2.5-7B-Instruct without backend-specific training. DeepSeek-V4-Flash performs skill evolution in all three backends, including skill generation, insight updating, and reflection generation as applicable. These model choices differ from those in the original SkillRL, ExpeL, and Reflexion studies and define the integrated experimental setting evaluated here. Within each backend, the task-execution and skill-evolution models are held fixed across all acquisition policies and ablations.

Table 6 reports selector settings shared by ALFWorld and WebShop. K is the number of latent profile dimensions, and d_f is the dimension of the frozen task-profile representation. λ regularizes the ridge head, the fixed experimental coefficient $\beta_t = 0.1$ scales its UCB confidence bonus, and γ_g controls how strongly the gap weights concentrate on lower estimated profile coordinates. The theoretical guarantee uses the calibrated radii in equation 2. H is the number of rounds between ridge-statistic restarts. N is the warmup length and T is the bandit horizon. The total number of evolution rounds is $N + T = 140$ for ALFWorld and $N + T = 1000$ for WebShop. w/o UCB sets $\beta_t = 0$ while retaining the profile-gap weights. w/o gap retains $\beta_t = 0.1$ and replaces $\widehat{\mathbf{g}}_t$ with $\mathbf{1}/K$.

C.2.1 Surrogate Reward and Practice Success-Rate Changes

Figure 3 relates the observed gap-weighted surrogate reward $\widehat{r}_t$ in equation 1 to the contemporaneous change in cumulative practice success rate. Let $y_t \in \{0, 1\}$ be the outcome of bandit-phase round t, and define $C_t = t^{-1} \sum_{i=1}^{t} y_i$, excluding warmup. The plotted vertical value is

$$\Delta C_t = C_t - C_{t-1} = \frac{1}{t}(y_t - C_{t-1}).$$

We plot rounds with both a preceding bandit-phase cumulative rate and a subsequent logged pre-decision profile. This excludes the first bandit round, for which C_{t-1} is undefined, and the final logged round, for which the next profile is unavailable. We additionally omit rounds with $\widehat{r}_t = 0$. The horizontal value is the observed surrogate reward formed from the profile difference after the outcome, before the backend memory update. It is distinct from a predicted UCB score or a directly observed change in true skill.

The six panels share horizontal and vertical ranges. Both axes use symmetric logarithmic scales, with linear regions $[-10^{-5}, 10^{-5}]$ for reward and $[-0.001, 0.001]$ for success-rate change. The vertical and horizontal reference lines mark zero reward and zero change, respectively. Thus, the upper-right and lower-left regions show agreement in the signs of surrogate feedback and the running-rate change, while the other two regions show disagreement. Color denotes the overall interaction round, including warmup. The figure shows the individual practice interactions, not held-out evaluation results.

Let ρ_{P} denote the empirical Pearson correlation computed from the untransformed plotted values. For SkillRL, ExpeL, and Shared Reflexion, respectively, ρ_{P} is 0.47, 0.03, and 0.64 on ALFWorld, and 0.57, 0.88, and 0.02 on WebShop. The associations are positive in all six runs but vary substantially, with two effectively near zero. Large positive changes in the cumulative rate occur at relatively few rounds. Because $\Delta C_t = (y_t - C_{t-1})/t$, earlier successful rounds can yield larger changes simply through the $1/t$ factor; these points do not establish unusually large skill updates. Since the backend memory update follows the observed outcome, these points cannot identify that update as the cause of the same round's success. Overall, the positive correlations provide a lim-

ited descriptive check on the reward's alignment with practice outcomes, not a causal test of task selection or held-out transfer.

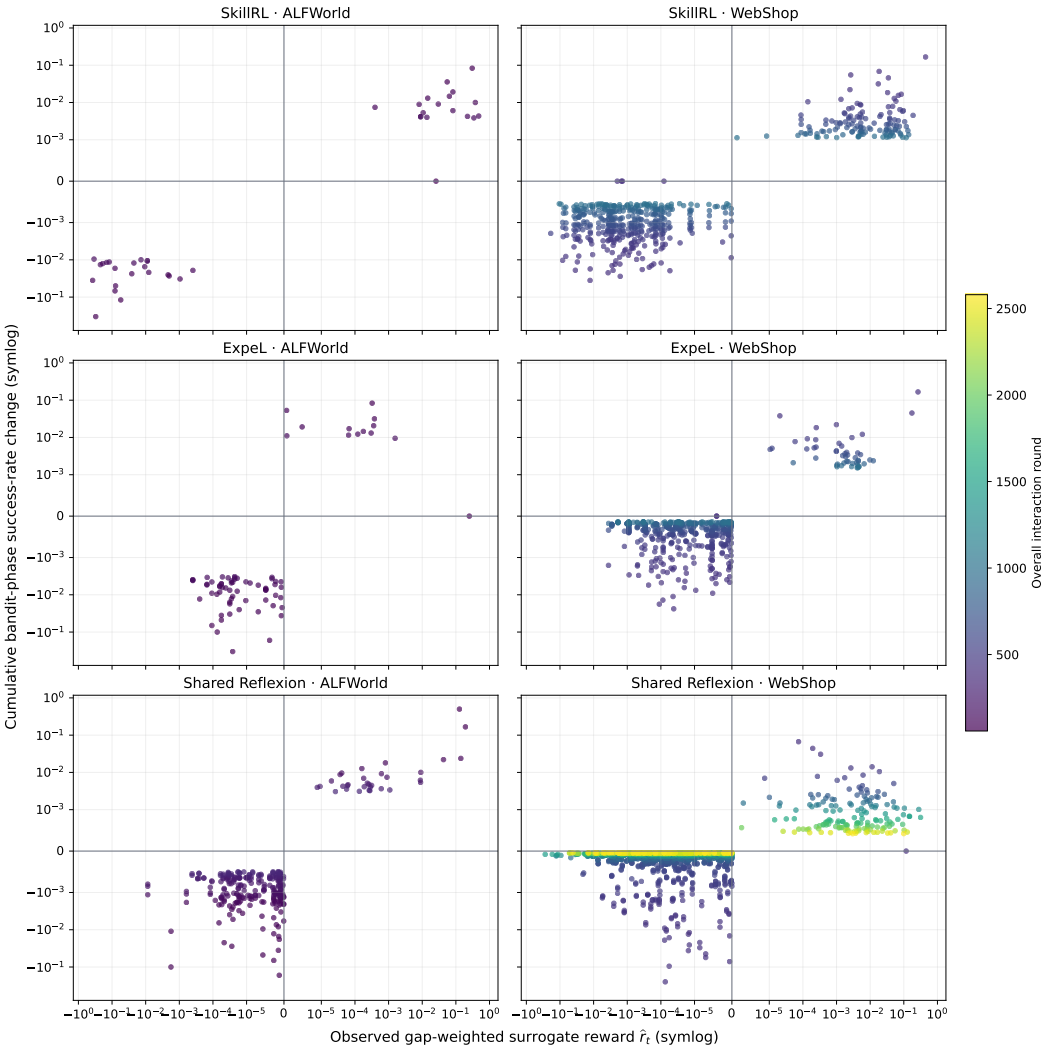


Figure 3: Observed gap-weighted surrogate reward versus the change in cumulative bandit-phase practice success rate. Rows show SkillRL, ExpeL, and Shared Reflexion, and columns show ALFWorld and WebShop.

Probe-based transfer analysis. The contemporaneous analysis above cannot by itself distinguish surrogate alignment with the current outcome from alignment with transferable improvement. We therefore conduct a separate probe-based analysis. Figure 4 shows 50 ALFWorld SkillRL experiments that compare surrogate reward with probe performance gains. Each experiment evaluates the same probe tasks before and after skill evolution. The probe uses a frozen skill state. Its trajectories and outcomes do not participate in skill evolution. All 50 displayed runs have positive mean surrogate reward. Probe success-rate gain is positive in 36 and zero in 14. These runs support using the surrogate as a reward signal for skill evolution.

Table 8 summarizes the 50 points. The variance is the sample variance across experiments.

Each experiment applies only one skill update. Thus, the small magnitude of $\widehat{r}$ is expected for a single update rather than evidence of negligible signal. Its small variance indicates that the surrogate-reward estimates are stable across the 50 experiments. The correspondingly small probe success-rate changes are consistent with the stability of the held-out probe set under a one-step update.

Table 8: Summary statistics for the 50 ALFWorld SkillRL probe-based transfer experiments.

Metric	Mean	Sample variance
Mean gap-weighted surrogate reward $\widehat{r}$	0.02788	1.47×10^{-6}
Probe success-rate gain	0.05600	2.11×10^{-3}

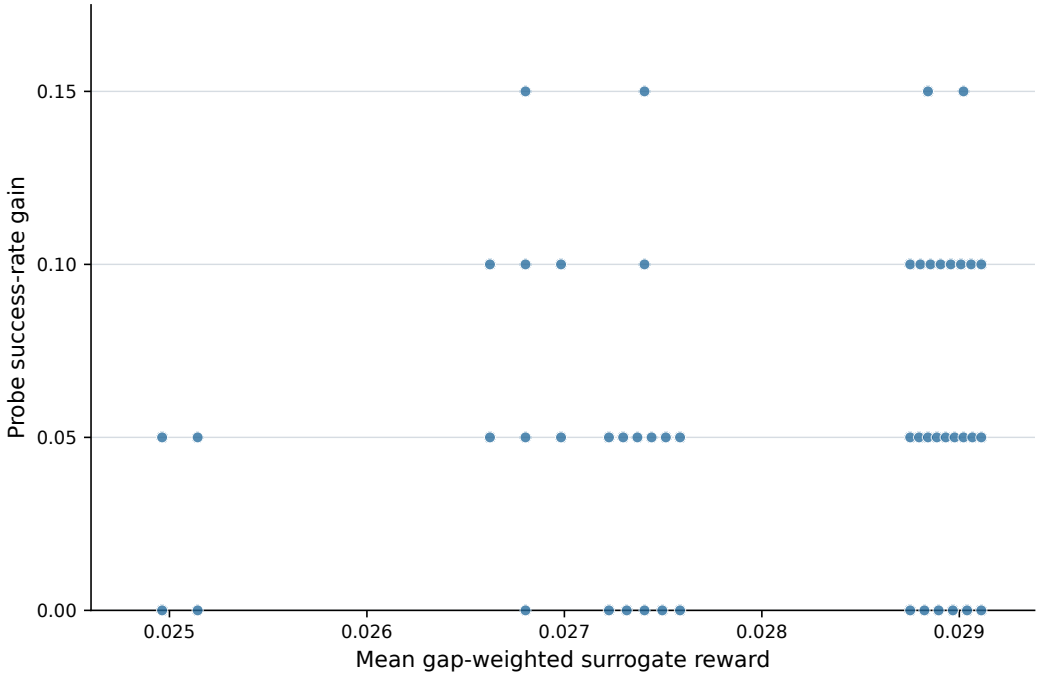


Figure 4: Mean gap-weighted surrogate reward versus probe success-rate gain for 50 ALFWorld SkillRL experiments. Each experiment evaluates the same probe tasks before and after skill evolution. Probe trajectories and outcomes are excluded from skill evolution. The displayed set contains 36 positive and 14 zero gains. A very small jitter is added to overlapping points for visual clarity.

C.2.2 BACKEND INTEGRATION

We retain failure-driven SkillBank expansion and skill retrieval for SkillRL (Xia et al., 2026). For ExpeL (Zhao et al., 2024), we retain trajectory-based reflection, successful-experience retrieval, and insight-bank updates. The integrated interface names its operations AGREE, REMOVE, EDIT, and ADD. For Shared Reflexion, we retain Reflexion's failure-to-corrective-plan feedback mechanism (Shinn et al., 2023). The remaining changes are backend-specific. Our SkillRL integration excludes the supervised fine-tuning and reinforcement-learning stages of the complete training pipeline. ExpeL's experience gathering and insight extraction are interleaved with online task acquisition rather than executed as separate stages. Shared Reflexion extends task-local episodic memory to a cross-task retrievable memory. Consequently, the experiments evaluate SPG-Select with the skill-evolution mechanisms of SkillRL, ExpeL, and Shared Reflexion, rather than exact reproductions of their complete original pipelines.

C.2.3 TOTAL EVALUATION API CALLS

Figure 5 reports total API calls over the complete held-out evaluation pool. The observed total for SPG-Select-R is approximately 8% to 42% lower than that for Uniform in five of the six backend-benchmark combinations. The exception is ExpeL on WebShop, for which SPG-Select-R uses approximately 22% more calls. Comparisons are made within each backend and benchmark since the environments and backend prompts induce different call scales.

C.3 PROMPT TEMPLATES AND A NEWLY GENERATED SKILL

Following the presentation convention of the SkillRL prompt appendix (Xia et al., 2026), we report each template in a titled box. The contents below are the templates used by our vendored SkillRL adapter rather than transcriptions of SkillRL's paper. Braced terms denote values filled at runtime. The first two prompts are invoked at every environment step after retrieval, whereas Prompt B.1 is invoked on a batch of failed trajectories to expand the skill bank.

Prompt A.1: ALFWorld Agent Execution with Retrieved Skills

System Prompt:
You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: {task_description}
Retrieved Relevant Experience
{retrieved_memories}
Current Task State
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}
You are now at step {current_step} and your current observation is: {current_observation}
Your admissible actions of the current situation are: [{admissible_actions}].
Now it is your turn to take an action. First reason step-by-step about the current situation. Enclose this reasoning within <think> and </think> tags. Then choose one admissible action for the current step and present it within <action> and </action> tags.

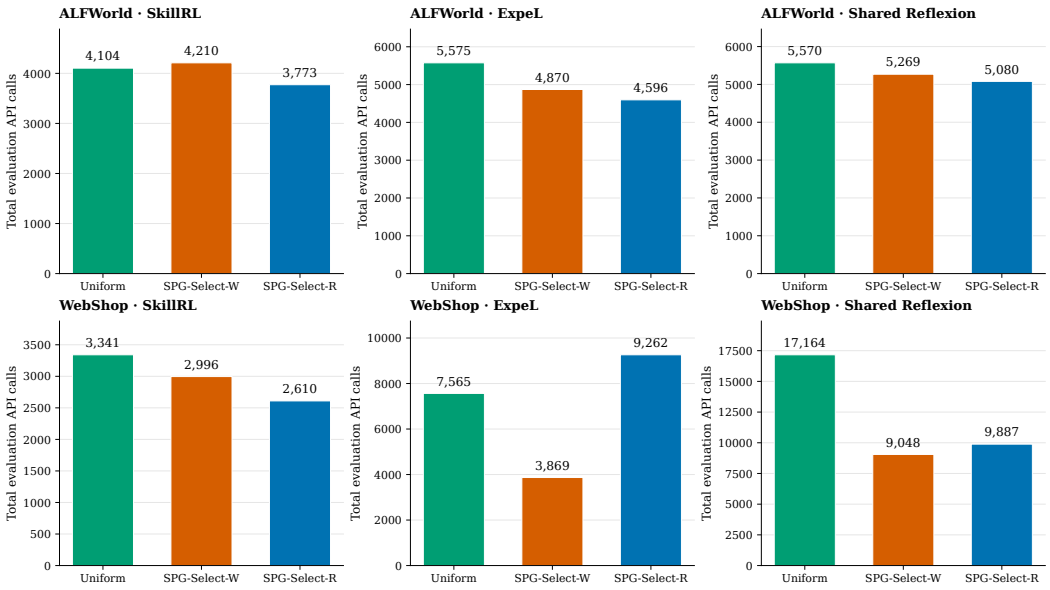


Figure 5: Total API calls over the complete held-out evaluation pool. Each bar aggregates the calls recorded for all 134 ALFWorld tasks or all 500 WebShop tasks under one task-acquisition policy. Comparisons are made within a backend and benchmark since the environments and backend prompts induce different call scales.

Prompt A.2: WebShop Agent Execution with Retrieved Skills

System Prompt:
You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: {task_description}.
Retrieved Relevant Experience
{retrieved_memories}
Current Task State
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}
You are now at step {current_step} and your current observation is: {current_observation}.
Your admissible actions of the current situation are: [{available_actions}].
Now it is your turn to take one action for the current step. First reason step-by-step about the current situation. Enclose this reasoning within <think> and </think> tags. Then choose one admissible action and present it within <action> and </action> tags. Inside <action>, copy exactly one native action from the admissible-action list, including its spelling and capitalization. Do not invent controls, abbreviate labels, or put the action in <think>.

Prompt B.1: Failure Reflection and New-Skill Generation

Analyze these failed agent trajectories and suggest NEW skills to add to the skill bank.
FAILED TRAJECTORIES:
{failure_examples}
EXISTING SKILL TITLES (avoid duplicating these):
{existing_titles}
Generate 1–{max_new_skills_per_update} NEW actionable skills that would help avoid these failures. Each skill must have: skill_id, title (3–5 words), principle (1–2 sentences), and when_to_apply.
Use skill IDs: {available_dynamic_skill_ids}.
Return ONLY a JSON array of skills, with no other text.
Example format:

```
[{"skill_id": "dyn_001", "title": "Verify Object Location First",
"principle": "Before attempting to pick up an object, verify its
current location by examining the environment.",
"when_to_apply": "When a task requires moving an object but its
location is uncertain"}]
```

Newly generated skill example. The following skill was generated for this appendix using the schema in Prompt B.1. It is an illustrative output, separate from the skills used in the reported experimental runs. At runtime, the updater replaces the illustrative identifier with the next unused dyn_NNN identifier.

Field	Newly generated value
skill_id	dyn_001
title	Require Observable State Change
principle	After an action intended to alter the world, compare the next observation with the pre-action state and treat the transition as unconfirmed unless a goal-relevant change is visible. If no change is observed, repair an unmet precondition or choose another admissible action instead of advancing the plan.
when_to_apply	After an open, take, place, heat, cool, clean, or variant-selection action whose success is required by a later step.

C.4 Qualitative Examples

This subsection presents successful held-out evaluation trajectories from the persisted logs of the main experiments.

C.4.1 Successful Held-Out Evaluation Cases

The cases below show every interaction step in each successful trajectory, with model thoughts omitted. Observations are abbreviated from the persisted evaluation logs for presentation. Ellipses shorten long room inventories, option lists, or product lists without omitting any interaction step.

Case 1: ALFWorld Warmed-Mug Placement (State Change and Delivery)

Task: Heat some mug and put it in the coffeemachine.

Skills	Systematic Exploration; Grab When Seen.
Avoid	Revisiting searched locations; ignoring visible targets.

Agent Trajectory with Retrieved Skill Guidance

Step 1	**Obs:** `-= Welcome to TextWorld, ALFRED! =-` You are in the middle of a room. You see cabinets 1–6, coffeemachine 1, countertops 1–3, drawers 1–3, fridge 1, garbagecan 1, microwave 1, shelves 1–3, sinkbasin 1, stoveburners 1–4, toaster 1, . . . **Action:** `go to cabinet 1`
Step 2	**Obs:** You arrive at cabinet 1. On cabinet 1, you see mug 1, plate 2, and plate 1. **Action:** `take mug 1 from cabinet 1`
Step 3	**Obs:** You pick up mug 1 from cabinet 1. **Action:** `go to microwave 1`
Step 4	**Obs:** You arrive at microwave 1. Microwave 1 is closed. **Action:** `open microwave 1`
Step 5	**Obs:** You open microwave 1. Microwave 1 is open; inside is egg 1. **Action:** `heat mug 1 with microwave 1`
Step 6	**Obs:** You heat mug 1 using microwave 1. **Action:** `go to coffeemachine 1`
Step 7	**Obs:** You arrive at coffeemachine 1. On coffeemachine 1, you see mug 2. **Action:** `move mug 1 to coffeemachine 1`

Result: SUCCESS — You move mug 1 to coffeemachine 1.

Figure 6: Complete seven-step logged trajectory of successful ALFWorld evaluation task 15.

Case 2: WebShop Brown Rice Cakes (Attribute and Price Constraints)

Task: Find a 12-pack of 9-ounce certified-organic, non-GMO, gluten-free brown rice cakes priced below $80.

Skills	Ground Constraints in Page Text; Click Exact Displayed Identifier.
Avoid	Unverified attribute assumptions; clicking non-controls.

Agent Trajectory with Retrieved Skill Guidance

Step 1	**Obs:** "I am looking for 9 ounce packs of 12 certified organic, non-GMO, and gluten-free organic brown rice cakes, and price lower than 80.00 dollars." `[Search]` **Action:** `search[organic brown rice cakes 9 oz 12 pack]`
Step 2	**Obs:** Page 1 (50 results): `[B0013JLZBU]` Cinnamon Toast, 9.5 oz, $100.00; `[B01GZI0JMY]` Honey Nut, 9.6 oz (pack of 12), gluten-free, USDA-certified organic, non-GMO verified, $56.60; `[B00M8YPW0O]` Lightly Salted, 8.5 oz (pack of 12), $68.79; . . . **Action:** `click[B01GZI0JMY]`
Step 3	**Obs:** Product page for Honey Nut, 9.6 oz (pack of 12), gluten-free, USDA-certified organic, non-GMO verified, $56.60. Available sizes include `[9 ounce (pack of 12)]`;... **Action:** `click[9 ounce (pack of 12)]`
Step 4	**Obs:** You have clicked `9 ounce (pack of 12)`. The refreshed product page shows the selected size, the same certified-organic, non-GMO, gluten-free product, price $56.60, and `[Buy Now]`. **Action:** `click[Buy Now]`

Result: SUCCESS — Environment score: 1.0.

Figure 7: Complete four-step logged trajectory of successful WebShop evaluation task 46.

D Proofs for Theoretical Analysis

This appendix states the assumptions and provides the supporting lemmas and complete proofs for Theorem 5.2 and Corollary D.8. Throughout, probabilities and expectations are conditional on the warmup outputs, and the assumptions below hold under this conditional law. Every epoch begins with a fresh ridge estimator, so no observation before the epoch start enters its design matrix or response statistic. The self-normalized construction follows Abbasi-Yadkori et al. (2011, Theorem 1), and we give the derivation needed for the vector-valued feedback and restarted design used here.

D.1 Assumptions

Assumption D.1. *There are predictable conditional-mean parameter matrices $\{\mathbf{W}_t^\star\}_{t=1}^T$, where $\mathbf{W}_t^\star \in \mathbb{R}^{d_f \times K}$ is $\mathcal{F}_{t-1}$-measurable, such that, for every $t \le T$ and $\tau \in \mathcal{T}$, $\widehat{\boldsymbol{\delta}}_t(\tau) = \mathbf{W}_t^{\star\top}\boldsymbol{\phi}_{t,\tau} + \boldsymbol{\varepsilon}_t(\tau)$, where $\boldsymbol{\varepsilon}_t(\tau) \in \mathbb{R}^K$ and $\mathbb{E}[\boldsymbol{\varepsilon}_t(\tau) \mid \mathcal{F}_{t-1}] = \mathbf{0}$.*

Assumption D.2. *The encoder $\widehat{\Phi}$ and MIRT item-parameter estimates $\{\widehat{\mathbf{a}}_\tau, \widehat{d}_\tau\}_{\tau\in\mathcal{T}}$ are fixed before the bandit phase, and every profile estimate used to construct a feature depends only on information available before that round. Thus, $\widehat{\mathbf{s}}_t$, its gap vector $\widehat{\mathbf{g}}_t$, and each feature $\boldsymbol{\phi}_{t,\tau}$ are $\mathcal{F}_{t-1}$-measurable. The encoder normalization ensures $\|\boldsymbol{\phi}_{t,\tau}\|_2 \le 1$. There are positive constants S and σ such that, for every $t \le T$, $\|\mathbf{W}_t^\star\|_F \le S$, where $\|\cdot\|_F$ denotes the Frobenius norm, and, for every $\tau \in \mathcal{T}$, every $\mathbf{v} \in \mathbb{R}^K$ with $\|\mathbf{v}\|_2 = 1$, and every $u \in \mathbb{R}$,*

$$\mathbb{E}\big[\exp\big(u\mathbf{v}^\top\boldsymbol{\varepsilon}_t(\tau)\big)\big|\mathcal{F}_{t-1}\big] \le \exp\left(\frac{\sigma^2u^2}{2}\right).$$

Definition D.3. The realized Frobenius path length of the predictable parameter sequence is $P_T = \sum_{t=1}^{T-1}\|\mathbf{W}_{t+1}^\star - \mathbf{W}_t^\star\|_F$. This quantity may be random and may depend on the policy.

By the triangle inequality and Assumption D.2, each increment has norm at most $2S$, so $0 \le P_T \le 2S(T-1)$ almost surely. The regret bound below retains P_T explicitly to quantify its dependence on parameter variation.

D.2 Lemmas for Theorem 5.2

Fix a restart period $H \in \{1, \dots, T\}$ and a failure probability $\rho \in (0, 1)$. The UCB confidence radius used in the main text is

$$\beta_t = \sigma \sqrt{2K \log\left(\frac{KT \det(\mathbf{A}_t)^{1/2}}{\rho \lambda^{d_f/2}} \right)} + \sqrt{\lambda} S, \qquad \beta = \max_{1 \le t \le T} \beta_t. \tag{2}$$

Here, $\mathbf{A}_t$ is the regularized design matrix of the current epoch. The derivation has three components. Lemma D.4 controls one scalar response coordinate through a self-normalized martingale inequality. Lemma D.6 applies this bound to the K coordinates of vector-valued feedback and takes a union bound over at most T restart epochs, producing the logarithmic factor in equation 2. The same lemma combines the coordinate-wise bounds in Frobenius norm, which yields the factor $\sqrt{K}$, and bounds the ridge regularization term by $\sqrt{\lambda} S$. The radii are positive because $\sigma, S, \lambda > 0$, $KT/\rho > 1$, and $\det(\mathbf{A}_t) \ge \lambda^{d_f}$. The following lemmas provide each step in full.

Fix an epoch $\mathcal{E} = \{t_0, \dots, t_0 + h - 1\}$, where $h \le H$. For $t \in \mathcal{E}$, ridge regression uses only $i \in \{t_0, \dots, t-1\}$. Write

$$P(\mathcal{E}_t) = \sum_{j=t_0+1}^{t} \|\mathbf{W}_j^\star - \mathbf{W}_{j-1}^\star\|_F \quad \text{and} \quad \mathbf{A}_{\mathcal{E}}^+ = \lambda \mathbf{I}_{d_f} + \sum_{i \in \mathcal{E}} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top.$$

Here, $P(\mathcal{E}_t)$ includes the change from $\mathbf{W}_{t-1}^\star$ to the current $\mathbf{W}_t^\star$, although $\mathcal{E}_t$ indexes observations only through $t-1$. All empty sums are zero, including at an epoch's first round. For any positive-definite matrix $\mathbf{Q}$, write $\|\mathbf{x}\|_{\mathbf{Q}} = \sqrt{\mathbf{x}^\top \mathbf{Q} \mathbf{x}}$.

Lemma D.4. *Let $\{\mathbf{x}_i, \xi_i\}_{i=t_0}^{t_0+h-1}$ be processes adapted to the bandit history, where $\mathbf{x}_i \in \mathbb{R}^{d_f}$ is $\mathcal{F}_{i-1}$-measurable, $\|\mathbf{x}_i\|_2 \le 1$, and $\xi_i \in \mathbb{R}$ is $\mathcal{F}_i$-measurable and conditionally σ-sub-Gaussian given $\mathcal{F}_{i-1}$. Put $\mathbf{S}_t = \sum_{i=t_0}^{t-1} \mathbf{x}_i \xi_i$ and $\mathbf{V}_t = \lambda \mathbf{I}_{d_f} + \sum_{i=t_0}^{t-1} \mathbf{x}_i \mathbf{x}_i^\top$. Then, for every $\chi \in (0, 1)$, with probability at least $1 - \chi$, the following inequality holds simultaneously for all $t \in \mathcal{E}$,*

$$\|\mathbf{S}_t\|_{\mathbf{V}_t^{-1}}^2 \le 2\sigma^2 \log\left(\frac{\det(\mathbf{V}_t)^{1/2}}{\chi \lambda^{d_f/2}} \right).$$

Proof. Fix $\boldsymbol{\omega} \in \mathbb{R}^{d_f}$. Conditional σ-sub-Gaussianity gives

$$\mathbb{E}\left[\exp\left(\boldsymbol{\omega}^\top \mathbf{x}_i \xi_i - \frac{\sigma^2}{2} (\boldsymbol{\omega}^\top \mathbf{x}_i)^2 \right) \middle| \mathcal{F}_{i-1} \right] \le 1.$$

Consequently,

$$M_t(\boldsymbol{\omega}) = \exp\left(\boldsymbol{\omega}^\top \mathbf{S}_t - \frac{\sigma^2}{2} \boldsymbol{\omega}^\top (\mathbf{V}_t - \lambda \mathbf{I}_{d_f}) \boldsymbol{\omega} \right)$$

is a nonnegative supermartingale with respect to $(\mathcal{F}_{t-1})_{t=t_0}^{t_0+h}$, with initial value one. The prefix sums and matrices are defined through $t = t_0 + h$ for this argument. Indeed, for each admissible t,

$$\begin{aligned} \mathbb{E}[M_{t+1}(\boldsymbol{\omega}) \mid \mathcal{F}_{t-1}] &= M_t(\boldsymbol{\omega}) \, \mathbb{E}\left[\exp\left(\boldsymbol{\omega}^\top \mathbf{x}_t \xi_t - \frac{\sigma^2}{2} (\boldsymbol{\omega}^\top \mathbf{x}_t)^2 \right) \middle| \mathcal{F}_{t-1} \right] \\ &\le M_t(\boldsymbol{\omega}), \end{aligned}$$

where the inequality is the preceding conditional moment-generating-function bound. Integrating with respect to the Gaussian density of $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, (\sigma^2 \lambda)^{-1} \mathbf{I}_{d_f})$ preserves the supermartingale prop-

erty by Tonelli's theorem. Completing the square gives

$$\begin{aligned}\widetilde{M}_t &= \left(\frac{\sigma^2\lambda}{2\pi}\right)^{d_f/2} \int_{\mathbb{R}^{d_f}} \exp\left(\boldsymbol{\omega}^\top \mathbf{S}_t - \frac{\sigma^2}{2}\boldsymbol{\omega}^\top \mathbf{V}_t \boldsymbol{\omega}\right) d\boldsymbol{\omega} \\ &= \left(\frac{\sigma^2\lambda}{2\pi}\right)^{d_f/2} \exp\left(\frac{\|\mathbf{S}_t\|^2_{\mathbf{V}_t^{-1}}}{2\sigma^2}\right) \int_{\mathbb{R}^{d_f}} \exp\left(-\frac{\sigma^2}{2}\left\|\boldsymbol{\omega} - \frac{\mathbf{V}_t^{-1}\mathbf{S}_t}{\sigma^2}\right\|^2_{\mathbf{V}_t}\right) d\boldsymbol{\omega} \\ &= \frac{\lambda^{d_f/2}}{\det(\mathbf{V}_t)^{1/2}} \exp\left(\frac{\|\mathbf{S}_t\|^2_{\mathbf{V}_t^{-1}}}{2\sigma^2}\right).\end{aligned}$$

The Gaussian integral is finite because $\mathbf{V}_t \succeq \lambda \mathbf{I}_{d_f} \succ 0$, and the mixture has initial value one. To obtain the maximal bound, let κ be the first $t \in \mathcal{E}$ with $\widetilde{M}_t \geq 1/\chi$, setting $\kappa = t_0 + h$ if no crossing occurs. This is a bounded stopping time for the shifted filtration. The bounded optional stopping theorem and nonnegativity give

$$1 \geq \mathbb{E}[\widetilde{M}_\kappa] \geq \chi^{-1} \Pr\left(\max_{t\in\mathcal{E}} \widetilde{M}_t \geq \chi^{-1}\right).$$

On the complementary event, take logarithms in the displayed expression for $\widetilde{M}_t$, multiply by $2\sigma^2 > 0$, and rearrange to obtain the claimed inequality for every $t \in \mathcal{E}$. □

Lemma D.5. *Under the feature bound in Assumption D.2, for $j \in \{t_0, \ldots, t-1\}$, let $\mathbf{C}_{t,j} = \sum_{i=t_0}^{j} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top$. Then $\|\mathbf{A}_t^{-1}\mathbf{C}_{t,j}\|_2 \leq \sqrt{d_f H/\lambda}$, where $\|\cdot\|_2$ for a matrix denotes its operator norm induced by the Euclidean norm.*

Proof. Following the matrix estimate in the proof of Zhao et al. (2020, Lemma 2), let $\mathbf{z}$ and $\widetilde{\mathbf{z}}$ be unit vectors in $\mathbb{R}^{d_f}$ attaining the bilinear characterization of $\|\mathbf{A}_t^{-1}\mathbf{C}_{t,j}\|_2$. By the Cauchy-Schwarz inequality in the $\mathbf{A}_t^{-1}$-inner product,

$$\begin{aligned}\left|\widetilde{\mathbf{z}}^\top \mathbf{A}_t^{-1}\mathbf{C}_{t,j}\mathbf{z}\right| &\leq \sum_{i=t_0}^{j} \|\widetilde{\mathbf{z}}\|_{\mathbf{A}_t^{-1}} \|\boldsymbol{\phi}_i\|_{\mathbf{A}_t^{-1}} \left|\boldsymbol{\phi}_i^\top \mathbf{z}\right| \\ &\leq \lambda^{-1/2} \sum_{i=t_0}^{j} \|\boldsymbol{\phi}_i\|_{\mathbf{A}_t^{-1}} \\ &\leq \lambda^{-1/2} \sqrt{H \sum_{i=t_0}^{j} \|\boldsymbol{\phi}_i\|^2_{\mathbf{A}_t^{-1}}} \\ &= \lambda^{-1/2} \sqrt{H \operatorname{tr}(\mathbf{A}_t^{-1}\mathbf{C}_{t,j})}.\end{aligned}$$

The second inequality uses $\mathbf{A}_t \succeq \lambda \mathbf{I}_{d_f}$, which yields $\|\widetilde{\mathbf{z}}\|_{\mathbf{A}_t^{-1}} \leq \lambda^{-1/2}$, together with $|\boldsymbol{\phi}_i^\top \mathbf{z}| \leq \|\boldsymbol{\phi}_i\|_2 \|\mathbf{z}\|_2 \leq 1$. The third inequality follows from the Cauchy-Schwarz inequality and uses at most H summands. Since $j \leq t-1$,

$$\mathbf{C}_{t,j} \preceq \sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top \preceq \mathbf{A}_t.$$

Congruence by $\mathbf{A}_t^{-1/2}$ gives $\mathbf{A}_t^{-1/2}\mathbf{C}_{t,j}\mathbf{A}_t^{-1/2} \preceq \mathbf{I}_{d_f}$. Taking traces and using cyclicity of the trace yields $\operatorname{tr}(\mathbf{A}_t^{-1}\mathbf{C}_{t,j}) \leq d_f$. Substitution in the preceding display proves the result. □

Lemma D.6. *Under Assumptions D.1 and D.2, with probability at least $1-\rho$, simultaneously for every restart epoch, every $t \in \mathcal{E}$, and every $\tau \in \mathcal{T}$,*

$$\left|\widehat{\mathbf{g}}_t^\top (\widehat{\mathbf{W}}_t - \mathbf{W}_t^\star)^\top \boldsymbol{\phi}_{t,\tau}\right| \leq \beta_t \|\widehat{\mathbf{g}}_t\|_2 \|\boldsymbol{\phi}_{t,\tau}\|_{\mathbf{A}_t^{-1}} + \sqrt{\frac{d_f H}{\lambda}} P(\mathcal{E}_t).$$

Proof. For the selected task at round i, Assumption D.1 gives $\widehat{\boldsymbol{\delta}}_i = \mathbf{W}_i^{\star\top}\boldsymbol{\phi}_i + \boldsymbol{\varepsilon}_i$, where $\boldsymbol{\varepsilon}_i = \boldsymbol{\varepsilon}_i(\tau_i)$. Hence

$$\mathbf{B}_t = \sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \widehat{\boldsymbol{\delta}}_i^\top = \sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top \mathbf{W}_i^\star + \sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\varepsilon}_i^\top.$$

Subtracting $\mathbf{W}_t^\star = \mathbf{A}_t^{-1}\mathbf{A}_t\mathbf{W}_t^\star$ from $\widehat{\mathbf{W}}_t = \mathbf{A}_t^{-1}\mathbf{B}_t$ therefore gives

$$\widehat{\mathbf{W}}_t - \mathbf{W}_t^\star = \mathbf{A}_t^{-1}\left[\sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top (\mathbf{W}_i^\star - \mathbf{W}_t^\star) + \sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\varepsilon}_i^\top - \lambda \mathbf{W}_t^\star\right].$$

For the drift term, telescope $\mathbf{W}_i^\star - \mathbf{W}_t^\star = \sum_{j=i+1}^{t}(\mathbf{W}_{j-1}^\star - \mathbf{W}_j^\star)$. Interchanging the finite sums gives

$$\begin{aligned}\sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top (\mathbf{W}_i^\star - \mathbf{W}_t^\star) &= \sum_{j=t_0+1}^{t} \left(\sum_{i=t_0}^{j-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top\right)(\mathbf{W}_{j-1}^\star - \mathbf{W}_j^\star) \\ &= \sum_{j=t_0+1}^{t} \mathbf{C}_{t,j-1}(\mathbf{W}_{j-1}^\star - \mathbf{W}_j^\star).\end{aligned}$$

Lemma D.5, the Frobenius-norm triangle inequality, and the definition of $P(\mathcal{E}_t)$ now give

$$\begin{aligned}\left\|\mathbf{A}_t^{-1}\sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top (\mathbf{W}_i^\star - \mathbf{W}_t^\star)\right\|_F &\le \sum_{j=t_0+1}^{t} \|\mathbf{A}_t^{-1}\mathbf{C}_{t,j-1}\|_2 \|\mathbf{W}_{j-1}^\star - \mathbf{W}_j^\star\|_F \\ &\le \sqrt{\frac{d_f H}{\lambda}} P(\mathcal{E}_t).\end{aligned}$$

For the noise term, let $\mathbf{Z}_t = \sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\varepsilon}_i^\top$. The deterministic selection rule makes τ_i and $\boldsymbol{\phi}_i$ $\mathcal{F}_{i-1}$-measurable. The observed feedback $\widehat{\boldsymbol{\delta}}_i$ is $\mathcal{F}_i$-measurable, so the residual $\boldsymbol{\varepsilon}_i = \widehat{\boldsymbol{\delta}}_i - \mathbf{W}_i^{\star\top}\boldsymbol{\phi}_i$ is also $\mathcal{F}_i$-measurable. Write $\varepsilon_{i,k}$ for coordinate k of $\boldsymbol{\varepsilon}_i$. For every coordinate k, Assumption D.2 with $\mathbf{v}$ equal to the k-th standard basis vector gives conditional σ-sub-Gaussianity of $\varepsilon_{i,k}(\tau)$ for every candidate τ. Using the measurability of the selected τ_i consequently gives

$$\mathbb{E}[\exp(u\varepsilon_{i,k}(\tau_i))|\mathcal{F}_{i-1}] = \sum_{\tau\in\mathcal{T}} \mathbb{I}\{\tau_i = \tau\}\mathbb{E}[\exp(u\varepsilon_{i,k}(\tau))|\mathcal{F}_{i-1}] \le \exp\left(\frac{\sigma^2 u^2}{2}\right).$$

Thus Lemma D.4 applies to every coordinate of $\mathbf{Z}_t$. Apply it to coordinate k with $\chi = \rho/(KT)$. There are at most T restart epochs and K coordinates, so a union bound gives an event of probability at least $1-\rho$ on which, simultaneously for all epochs, times, and coordinates,

$$\left\|\mathbf{A}_t^{-1/2}\sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \varepsilon_{i,k}\right\|_2^2 \le 2\sigma^2 \log\left(\frac{KT\det(\mathbf{A}_t)^{1/2}}{\rho\lambda^{d_f/2}}\right).$$

Summing over $k = 1, \ldots, K$ and taking square roots yields

$$\|\mathbf{A}_t^{-1/2}\mathbf{Z}_t\|_F \le \sigma\sqrt{2K\log\left(\frac{KT\det(\mathbf{A}_t)^{1/2}}{\rho\lambda^{d_f/2}}\right)}.$$

Since $\mathbf{A}_t \succeq \lambda\mathbf{I}_{d_f}$, we have $\mathbf{A}_t^{-1} \preceq \lambda^{-1}\mathbf{I}_{d_f}$, and hence

$$\|\lambda\mathbf{A}_t^{-1/2}\mathbf{W}_t^\star\|_F \le \sqrt{\lambda}\|\mathbf{W}_t^\star\|_F \le \sqrt{\lambda}S.$$

Set $\mathbf{R}_t = \mathbf{A}_t^{-1/2}\mathbf{Z}_t - \lambda\mathbf{A}_t^{-1/2}\mathbf{W}_t^\star$. The preceding two displays and the triangle inequality give $\|\mathbf{R}_t\|_F \le \beta_t$. For any τ,

$$\widehat{\mathbf{g}}_t^\top\left(\mathbf{A}_t^{-1}\mathbf{Z}_t - \lambda\mathbf{A}_t^{-1}\mathbf{W}_t^\star\right)^\top \boldsymbol{\phi}_{t,\tau} = \widehat{\mathbf{g}}_t^\top \mathbf{R}_t^\top \mathbf{A}_t^{-1/2}\boldsymbol{\phi}_{t,\tau}.$$

The Cauchy-Schwarz inequality for the Frobenius inner product bounds the absolute value by $\|\mathbf{R}_t\|_F\|\widehat{\mathbf{g}}_t\|_2\|\boldsymbol{\phi}_{t,\tau}\|_{\mathbf{A}_t^{-1}}$, which is at most the statistical term in the lemma. Finally, define the drift matrix

$$\mathbf{G}_t = \mathbf{A}_t^{-1}\sum_{i=t_0}^{t-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\top(\mathbf{W}_i^\star - \mathbf{W}_t^\star).$$

Then

$$\left|\widehat{\mathbf{g}}_t^\top\mathbf{G}_t^\top\boldsymbol{\phi}_{t,\tau}\right| \le \|\mathbf{G}_t\|_F\|\widehat{\mathbf{g}}_t\|_2\|\boldsymbol{\phi}_{t,\tau}\|_2 \le \sqrt{\frac{d_f H}{\lambda}}P(\mathcal{E}_t),$$

where $\|\widehat{\mathbf{g}}_t\|_2 \le \|\widehat{\mathbf{g}}_t\|_1 = 1$ follows from the softmax definition and $\|\boldsymbol{\phi}_{t,\tau}\|_2 \le 1$ follows from Assumption D.2. Combining the drift, noise, and regularization bounds proves the result. □

Lemma D.7. *Under the feature bound in Assumption D.2, for every epoch $\mathcal{E}$,*

$$\sum_{t\in\mathcal{E}}\|\boldsymbol{\phi}_{t,\tau_t}\|_{\mathbf{A}_t^{-1}} \le \sqrt{\frac{(\lambda+1)d_f h}{\lambda}\log\left(1+\frac{h}{\lambda d_f}\right)}.$$

Moreover, its terminal design matrix satisfies

$$\frac{\det(\mathbf{A}_{\mathcal{E}}^+)}{\det(\lambda\mathbf{I}_{d_f})} \le \left(1+\frac{h}{\lambda d_f}\right)^{d_f}.$$

Proof. Put $u_t = \|\boldsymbol{\phi}_{t,\tau_t}\|^2_{\mathbf{A}_t^{-1}}$. The matrix determinant lemma gives

$$\det(\mathbf{A}_{t+1}) = \det\big(\mathbf{A}_t + \boldsymbol{\phi}_{t,\tau_t}\boldsymbol{\phi}_{t,\tau_t}^\top\big) = \det(\mathbf{A}_t)(1+u_t)$$

before any reset. Since $\mathbf{A}_t \succeq \lambda\mathbf{I}_{d_f}$ and $\|\boldsymbol{\phi}_{t,\tau_t}\|_2 \le 1$,

$$0 \le u_t \le \lambda^{-1}\|\boldsymbol{\phi}_{t,\tau_t}\|_2^2 \le \frac{1}{\lambda}.$$

For $u \ge 0$, the bound $(1+x)^{-1} \ge (1+u)^{-1}$ on $x \in [0,u]$ gives $\log(1+u) = \int_0^u (1+x)^{-1}\,dx \ge u/(1+u)$. Applying this with $u = u_t$, and then using $1+u_t \le 1+1/\lambda$, gives

$$u_t \le \left(1+\frac{1}{\lambda}\right)\log(1+u_t) = \frac{\lambda+1}{\lambda}\log(1+u_t).$$

Summing, telescoping the determinant identity, and applying the Cauchy-Schwarz inequality yield

$$\begin{aligned}\sum_{t\in\mathcal{E}}\sqrt{u_t} &\le \sqrt{h\sum_{t\in\mathcal{E}}u_t}\\ &\le \sqrt{\frac{(\lambda+1)h}{\lambda}\sum_{t\in\mathcal{E}}\log(1+u_t)}\\ &= \sqrt{\frac{(\lambda+1)h}{\lambda}\log\left(\frac{\det(\mathbf{A}_{\mathcal{E}}^+)}{\det(\lambda\mathbf{I}_{d_f})}\right)}.\end{aligned}$$

Here, $\mathbf{A}_{\mathcal{E}}^+$ is the terminal design matrix of the epoch, after its final observation and before any subsequent reset. To bound its determinant, let $\nu_1,\dots,\nu_{d_f} \ge 0$ be the eigenvalues of $\lambda^{-1}\sum_{i\in\mathcal{E}}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^\top$. The arithmetic-geometric mean inequality and $\sum_{j=1}^{d_f}\nu_j = \lambda^{-1}\sum_{i\in\mathcal{E}}\|\boldsymbol{\phi}_i\|_2^2 \le h/\lambda$ give

$$\frac{\det(\mathbf{A}_{\mathcal{E}}^+)}{\det(\lambda\mathbf{I}_{d_f})} = \prod_{j=1}^{d_f}(1+\nu_j) \le \left(1+\frac{1}{d_f}\sum_{j=1}^{d_f}\nu_j\right)^{d_f} \le \left(1+\frac{h}{\lambda d_f}\right)^{d_f}.$$

Substituting this inequality into the preceding display proves the result. □

D.3 Proof of Theorem 5.2

Proof. Write $\widehat{q}_t(\tau) = \widehat{\mathbf{g}}_t^\top \widehat{\mathbf{W}}_t^\top \boldsymbol{\phi}_{t,\tau}$, $w_t(\tau) = \|\boldsymbol{\phi}_{t,\tau}\|_{\mathbf{A}_t^{-1}}$, and $D_t = \sqrt{d_f H/\lambda} P(\mathcal{E}_t)$. Assumption D.1 and conditional mean-zero noise imply $q_t(\tau) = \widehat{\mathbf{g}}_t^\top \mathbf{W}_t^{\star\top} \boldsymbol{\phi}_{t,\tau}$. Thus, the event in Lemma D.6 implies, for every candidate τ,

$$q_t(\tau) \le \widehat{q}_t(\tau) + \beta_t \|\widehat{\mathbf{g}}_t\|_2 w_t(\tau) + D_t, \qquad q_t(\tau) \ge \widehat{q}_t(\tau) - \beta_t \|\widehat{\mathbf{g}}_t\|_2 w_t(\tau) - D_t,$$

where the first and second inequalities are respectively the upper and lower sides of that lemma. Algorithm 1 selects an action maximizing $\widehat{q}_t(\tau) + \beta_t \|\widehat{\mathbf{g}}_t\|_2 w_t(\tau)$. Applying the upper inequality to τ_t^*, UCB optimality, and then the lower inequality to τ_t, in this order, gives

$$\begin{aligned} q_t(\tau_t^*) - q_t(\tau_t) &\le \widehat{q}_t(\tau_t^*) + \beta_t \|\widehat{\mathbf{g}}_t\|_2 w_t(\tau_t^*) + D_t - q_t(\tau_t) \\ &\le \widehat{q}_t(\tau_t) + \beta_t \|\widehat{\mathbf{g}}_t\|_2 w_t(\tau_t) + D_t - q_t(\tau_t) \\ &\le 2\beta_t \|\widehat{\mathbf{g}}_t\|_2 w_t(\tau_t) + 2D_t \\ &\le 2\beta_t w_t(\tau_t) + 2D_t. \end{aligned}$$

The final inequality uses $\|\widehat{\mathbf{g}}_t\|_2 \le 1$. For an epoch $\mathcal{E} = \{t_0, \dots, t_0 + h - 1\}$, define $P(\mathcal{E}) = \sum_{j=t_0+1}^{t_0+h-1} \|\mathbf{W}_j^\star - \mathbf{W}_{j-1}^\star\|_F$. Since $\beta_t \le \beta$, Lemma D.7 and $h \le H$ give

$$\sum_{t\in\mathcal{E}} 2\beta_t w_t(\tau_t) \le 2\beta \sqrt{\frac{(\lambda+1)d_f h}{\lambda} \log\!\left(1 + \frac{h}{\lambda d_f}\right)} \le 2\beta \sqrt{\frac{(\lambda+1)d_f H}{\lambda} \log\!\left(1 + \frac{H}{\lambda d_f}\right)}.$$

Moreover, expanding the definition of the local path length and interchanging the finite sums yield

$$\begin{aligned} \sum_{t\in\mathcal{E}} P(\mathcal{E}_t) &= \sum_{t=t_0}^{t_0+h-1} \sum_{j=t_0+1}^{t} \|\mathbf{W}_j^\star - \mathbf{W}_{j-1}^\star\|_F \\ &= \sum_{j=t_0+1}^{t_0+h-1} (t_0 + h - j) \|\mathbf{W}_j^\star - \mathbf{W}_{j-1}^\star\|_F \\ &\le H \sum_{j=t_0+1}^{t_0+h-1} \|\mathbf{W}_j^\star - \mathbf{W}_{j-1}^\star\|_F = H P(\mathcal{E}), \end{aligned}$$

where the inequality uses $t_0 + h - j \le h - 1 \le H$. Consequently,

$$\sum_{t\in\mathcal{E}} 2D_t = 2\sqrt{\frac{d_f H}{\lambda}} \sum_{t\in\mathcal{E}} P(\mathcal{E}_t) \le 2\sqrt{\frac{d_f}{\lambda}} H^{3/2} P(\mathcal{E}).$$

There are exactly $\lceil T/H \rceil$ epochs, with the last possibly shorter than H. Their within-epoch variation sums satisfy $\sum_{\mathcal{E}} P(\mathcal{E}) \le P_T$, since each increment $\|\mathbf{W}_j^\star - \mathbf{W}_{j-1}^\star\|_F$ can occur in at most one within-epoch path length. Summing the preceding estimation and drift bounds over the epochs gives

$$R_T \le 2\beta \left\lceil \frac{T}{H} \right\rceil \sqrt{\frac{(\lambda+1)d_f H}{\lambda} \log\!\left(1 + \frac{H}{\lambda d_f}\right)} + 2\sqrt{\frac{d_f}{\lambda}} H^{3/2} P_T,$$

which is the claimed high-probability conclusion. □

D.4 Oracle Choice of the Restart Period

For restart-period selection, define the horizon-uniform UCB radius

$$\beta^{\mathrm{u}} = \sigma \sqrt{2K \left[\log\!\left(\frac{KT}{\rho}\right) + \frac{d_f}{2} \log\!\left(1 + \frac{T}{\lambda d_f}\right) \right]} + \sqrt{\lambda} S. \tag{3}$$

We next verify the deterministic comparison between the time-specific and horizon-uniform radii. Fix $t \in \{1, \dots, T\}$, and let $\mathcal{E} = \{t_0, \dots, t_0 + h - 1\}$ be the epoch containing t. By the definitions of the prefix and terminal design matrices,

$$\mathbf{A}_t = \lambda \mathbf{I}_{d_f} + \sum_{i=t_0}^{t-1} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top \preceq \lambda \mathbf{I}_{d_f} + \sum_{i\in\mathcal{E}} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top = \mathbf{A}_{\mathcal{E}}^+.$$

The matrix $\mathbf{A}_t^{-1/2}\mathbf{A}_{\mathcal{E}}^{+}\mathbf{A}_t^{-1/2} \succeq \mathbf{I}_{d_f}$ has all eigenvalues at least one. Thus,

$$\frac{\det(\mathbf{A}_{\mathcal{E}}^{+})}{\det(\mathbf{A}_t)} = \det\Big(\mathbf{A}_t^{-1/2}\mathbf{A}_{\mathcal{E}}^{+}\mathbf{A}_t^{-1/2}\Big) \geq 1.$$

Applying this observation and Lemma D.7 gives

$$\frac{\det(\mathbf{A}_t)^{1/2}}{\lambda^{d_f/2}} \leq \frac{\det(\mathbf{A}_{\mathcal{E}}^{+})^{1/2}}{\lambda^{d_f/2}} \leq \left(1+\frac{h}{\lambda d_f}\right)^{d_f/2} \leq \left(1+\frac{T}{\lambda d_f}\right)^{d_f/2},$$

where the last inequality uses $h \leq H \leq T$. Consequently,

$$\begin{aligned}\beta_t &\leq \sigma\sqrt{2K\left[\log\left(\frac{KT}{\rho}\right)+\frac{d_f}{2}\log\left(1+\frac{T}{\lambda d_f}\right)\right]}+\sqrt{\lambda}S \\ &= \beta^{\mathrm{u}}.\end{aligned}$$

Since this holds for every $t \in \{1,\dots,T\}$, the definition $\beta = \max_{1\leq t\leq T}\beta_t$ yields $\beta \leq \beta^{\mathrm{u}}$.

Corollary D.8. *Fix the warmup outputs and the parameters of Theorem 5.2. Let $V_T \geq 0$ be a deterministic budget specified before the bandit phase. Suppose that the parameter path under the policy defined below satisfies $P_T \leq V_T$ almost surely. For $V_T > 0$, define*

$$L = (\lambda+1)\log\left(1+\frac{T}{\lambda d_f}\right), \qquad H^{\circ} = \left(\frac{\beta^{\mathrm{u}}T\sqrt{L}}{V_T}\right)^{1/2}, \qquad H^{*} = \min\{T, \max\{1, \lceil H^{\circ}\rceil\}\}.$$

Run Algorithm 1 with the fixed period $H = H^{*}$ and the radii in equation 2. With probability at least $1-\rho$, conditional on the warmup outputs,

$$R_T \leq \min\left\{T,\ 2\beta^{\mathrm{u}}\left\lceil\frac{T}{H^{*}}\right\rceil\sqrt{\frac{(\lambda+1)d_f H^{*}}{\lambda}\log\left(1+\frac{H^{*}}{\lambda d_f}\right)}+2\sqrt{\frac{d_f}{\lambda}}(H^{*})^{3/2}V_T\right\}. \tag{4}$$

If $1 \leq H^{\circ} \leq T$, the same event gives

$$R_T \leq (4+2^{5/2})\sqrt{\frac{d_f}{\lambda}}(\beta^{\mathrm{u}}T)^{3/4}L^{3/8}V_T^{1/4}.$$

When $K, \lambda, \lambda^{-1}, S, \sigma, \rho^{-1}$ are fixed independently of d_f and T, this simplifies to

$$R_T = \widetilde{\mathcal{O}}\Big(d_f^{7/8}T^{3/4}V_T^{1/4}\Big).$$

For $H^{\circ} > T$, the period is $H^{*} = T$, and the bound is at most $4\beta^{\mathrm{u}}\sqrt{d_f LT/\lambda}$. For $H^{\circ} < 1$, the period is $H^{*} = 1$, and equation 4 remains valid without the interior-rate simplification. Separately, if $V_T = 0$, use $H = T$. Then

$$R_T \leq 2\beta^{\mathrm{u}}\sqrt{\frac{(\lambda+1)d_f T}{\lambda}\log\left(1+\frac{T}{\lambda d_f}\right)},$$

which is $\widetilde{\mathcal{O}}(d_f\sqrt{T})$ in the same fixed-parameter regime.

Proof. The radius β^{u} and budget V_T are fixed before the bandit phase, so H^{*} is a nonanticipating integer in $\{1,\dots,T\}$. Apply Theorem 5.2 to this period, and intersect its confidence event with the probability-one event $P_T \leq V_T$. Substituting $P_T \leq V_T$ and $\beta \leq \beta^{\mathrm{u}}$ yields the second entry of the minimum in equation 4.

For the first entry, every candidate's potential next profile lies in $[0,1]^K$ and is compared with the same current profile $\widehat{\mathbf{s}}_t$. Therefore, the coordinatewise bounds on its temporal difference and their conditional expectations give

$$-\widehat{\mathbf{s}}_t \leq \boldsymbol{\mu}_t(\tau) \leq \mathbf{1}-\widehat{\mathbf{s}}_t.$$

The gap weights are nonnegative and sum to one, so

$$-\widehat{\mathbf{g}}_t^{\top}\widehat{\mathbf{s}}_t \leq q_t(\tau) \leq 1-\widehat{\mathbf{g}}_t^{\top}\widehat{\mathbf{s}}_t.$$

This interval has length one and is common to all candidates. Hence $0 \le q_t(\tau_t^*) - q_t(\tau_t) \le 1$ and $R_T \le T$ deterministically.

For $1 \le H^* \le T$, use $\lceil T/H^* \rceil \le 2T/H^*$ and $\log(1 + H^*/(\lambda d_f)) \le \log(1 + T/(\lambda d_f))$. The second entry of the minimum is therefore at most

$$\sqrt{\frac{d_f}{\lambda}} \left[\frac{4\beta^{\mathrm{u}} T\sqrt{L}}{\sqrt{H^*}} + 2(H^*)^{3/2} V_T \right].$$

The unrounded choice balances the two terms without their numerical constants, since

$$\frac{\beta^{\mathrm{u}} T\sqrt{L}}{\sqrt{H^\circ}} = (\beta^{\mathrm{u}} T\sqrt{L})^{3/4} V_T^{1/4} = V_T (H^\circ)^{3/2}.$$

If $1 \le H^\circ \le T$, then $H^* = \lceil H^\circ \rceil$ and $H^\circ \le H^* \le 2H^\circ$. Relative to H°, this rounding does not increase the estimation term and increases the drift term by at most a factor of $2^{3/2}$. The preceding bound becomes

$$R_T \le (4 + 2 \cdot 2^{3/2}) \sqrt{\frac{d_f}{\lambda}} (\beta^{\mathrm{u}} T\sqrt{L})^{3/4} V_T^{1/4},$$

which proves the stated constant and parameter dependence. Under the fixed-parameter regime in the corollary, equation 3 gives

$$\beta^{\mathrm{u}} = \sigma \sqrt{2K \left[\log\left(\frac{KT}{\rho}\right) + \frac{d_f}{2} \log\left(1 + \frac{T}{\lambda d_f}\right) \right]} + \sqrt{\lambda} S = \widetilde{\mathcal{O}}(\sqrt{d_f}),$$

while L is logarithmic in d_f, T. Thus the dimension factor is $d_f^{1/2} (d_f^{1/2})^{3/4} = d_f^{7/8}$, giving the interior rate.

If $H^\circ > T$, its definition implies $V_T < \beta^{\mathrm{u}}\sqrt{L}/T$. In equation 4, set $H^* = T$ to obtain

$$\begin{aligned} R_T &\le 2\beta^{\mathrm{u}} \sqrt{\frac{d_f LT}{\lambda}} + 2\sqrt{\frac{d_f}{\lambda}} T^{3/2} V_T \\ &\le 4\beta^{\mathrm{u}} \sqrt{\frac{d_f LT}{\lambda}}. \end{aligned}$$

If $H^\circ < 1$, clipping sets $H^* = 1$, already covered by the finite-sample bound. Finally, $V_T = 0$ implies $P_T = 0$ almost surely. Substituting $H = T$ into Theorem 5.2 and using $\beta \le \beta^{\mathrm{u}}$ proves the stationary bound. Its factors have orders $\widetilde{\mathcal{O}}(\sqrt{d_f})$ and $\widetilde{\mathcal{O}}(\sqrt{d_f T})$, respectively, so their product is $\widetilde{\mathcal{O}}(d_f\sqrt{T})$. □

The minimum in equation 4 makes explicit when the analysis improves on the trivial bound. For fixed $d_f, K, \lambda, S, \sigma, \rho$, the interior expression is $\mathcal{O}(T^{3/4} V_T^{1/4} (\log T)^{3/4})$; thus $V_T (\log T)^3 = o(T)$ is sufficient for a sublinear bound in that regime. A finite path length alone does not guarantee sublinear regret.