BELLMAN-CENTRIC LEARNING: NEAR-OPTIMAL REGRET FOR LINEAR BANDITS WITH MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

We study linear bandits with memory, where past actions induce endogenous non-stationarity through an arbitrary known, bounded matrix-valued memory map. To trade off exploration and exploitation while accounting for the memory dynamics, we develop RSM-LinUCB, a Bellman-centric algorithm that learns as in linear bandits and plans as in reinforcement learning. This design admits a novel regret decomposition which separates the memory-induced error from the cumulative reward estimation error along the learner's trajectory. We prove a high-probability regret bound of $\widetilde{O}\big(dRS(M+1)+\sigma d\sqrt{T}\big)$, where T is the learning horizon, d is the parameter dimension, M is the memory length, R and S bound the memory-map operator norm and reward-parameter norm, respectively, and σ is the sub-Gaussian noise scale. Our results reveal that the multiplicative memory–horizon coupling in prior bounds is not intrinsic: memory only contributes an additive cost, up to logarithmic factors. We also prove a matching minimax lower bound, establishing near-optimality. We further extend the algorithm to generalized linear rewards, preserving this separation with near-optimal memory and leading statistical dependence. Our algorithms outperform the baselines in numerical experiments on synthetic instances and semi-synthetic KV- and semantic-cache tasks.

1 INTRODUCTION

Linear bandits (Abbasi-Yadkori et al., 2011; Hao et al., 2020) are a widely studied framework for sequential decision-making that generalizes multi-armed bandits (Bubeck & Cesa-Bianchi, 2012; Auer et al., 2002) by modeling expected rewards as linear functions of action features. Recent work (Wu et al., 2018; Chen et al., 2019; Luo et al., 2018; Di Benedetto et al., 2020) extends linear bandits to the non-stationary setting, where the unknown reward parameter may change over time. Such temporal variation arises in applications including large language model (LLM) serving (Taberner-Miller, 2026; Lee & Wong, 2026), recommendation systems (Ghoorchian et al., 2024; Xu et al., 2020), and dynamic pricing (Zhao et al., 2026). In these environments, the learner must estimate rewards from noisy feedback while adapting to changes in the underlying environment.

Early work on non-stationary bandits focuses on exogenous non-stationarity, where the environment changes independently of the learner's actions (Garivier & Moulines, 2011; Besbes et al., 2014; Karnin & Anava, 2016). In many applications, however, the learner's recent decisions themselves influence subsequent rewards, giving rise to endogenous non-stationarity. One concrete example is key–value (KV) caching in LLM serving with a fixed retention window (Qin et al., 2025). Refreshing a prefix makes its KV tensors available for reuse in the current and subsequent rounds. The benefit of a refresh therefore depends on earlier decisions: refreshing an uncached prefix can enable reuse for additional requests, whereas refreshing an already cached prefix extends its availability. Related memory effects arise in model or adapter routing (Fu et al., 2024), where recent pinning and routing decisions determine which models or adapters remain warm. In recommendation and advertising (Warlop et al., 2018; Agrawal & Tulabandula, 2020), recent exposures modify users' responses through priming, boredom, or wear-out effects; for example, repeatedly displaying similar advertisements may cause user fatigue and reduce subsequent rewards. In dynamic pricing, past prices affect subsequent revenues through finite-memory reference price effects (den Boer & Keskin, 2022) or through their impact on resource occupancy and availability (Jia et al., 2024).

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.

Recent work on linear bandits with memory (LBM) (Clerici et al., 2023; Ann et al., 2026) studies one specific endogenous non-stationarity setting: the most recent M actions determine a known memory matrix $A(\cdot)$ that transforms a fixed unknown reward parameter. Their algorithms divide the learning horizon into episodes with fixed length and select an entire sequence of actions at the start of each episode by maximizing the optimistic reward objective. The best known regret bound with this approach is $\widetilde{O}\big(RS\sqrt{dMT} + \sigma \max\{1, S\}\sqrt{d \max\{d, R^2\}T}\big)$, where T is the horizon, M is the memory length, d is the reward-parameter dimension, σ is the sub-Gaussian noise, and R and S bound the operator norm of $A(\cdot)$ and the Euclidean norm of $\theta^\star$, respectively. The first regret term couples memory length and horizon through $\sqrt{MT}$, while the second introduces additional dependence on R and S relative to the usual $\widetilde{O}(\sigma d\sqrt{T})$ statistical term. A similar memory–horizon term appears in the extension to generalized linear bandits with memory (GLBM (Ann et al., 2026)). At the meantime, the existing literature does not yet provide minimax regret lower bounds that quantify the cost of memory. Thus, it remained an open question: is the multiplicative memory–horizon dependence intrinsic, or can near-optimal regret be achieved with an additive memory cost?

Memory-dependent rewards in these LBM and GLBM settings pose two main challenges. First, policy updates must account for persistent memory. Although new observations allow the learner to improve the policy, each updated policy inherits a memory state shaped by earlier decisions. Frequent updates may therefore accumulate memory-induced error, requiring a tradeoff between incorporating new information and maintaining policy stability. Second, learning must account for the long-term effects of actions. An action determines not only the immediate reward but also the memory state governing future rewards. The learning agent must therefore account for these long-term effects while learning the unknown reward parameter. Our key observation is that the most recent M actions in LBM naturally form a state with known and deterministic transitions, as in reinforcement learning (RL), while rewards are governed by a shared unknown parameter, as in linear bandits. Neither view alone suffices: a myopic bandit approach neglects future memory effects and can incur linear regret, whereas a direct tabular RL algorithm does not exploit the shared reward structure and yields a regret with exponential dependence on the memory length. We therefore ask:

> *Can we exploit this structure to learn from bandit feedback while planning over memory states as in RL, and achieve near-optimal regret?*

We develop algorithms and regret guarantees for LBM and GLBM. Our contributions are as follows.

- **We develop a Bellman-centric algorithm, RSM-LinUCB (Algorithm 1), in Section 3.** It constructs optimistic reward estimates using linear-bandit confidence sets and uses Bellman planning over memory states to account for the long-term effects of actions. To control the memory-induced costs of policy updates, RSM-LinUCB adopts determinant-triggered rare switching (Abbasi-Yadkori et al., 2011): it updates optimistic reward estimates and policy only when the determinant of the information matrix doubles. We further develop RSM-GLMUCB for GLBM in Section 5, extending optimistic reward estimation, Bellman planning, and rare policy updates to generalized linear rewards.
- **We establish near-optimal regret guarantees for RSM-LinUCB in Section 4.** Our analysis admits a Bellman regret decomposition that separates memory-induced boundary costs from cumulative reward-estimation error along the learner's trajectory. Specifically, Theorem 1 gives an upper bound of $\widetilde{O}\big(dRS(M+1) + \sigma d\sqrt{T}\big)$. Compared with prior bounds, this replaces multiplicative memory–horizon dependence with an additive memory cost, while removing polynomial dependence on R and S from the leading statistical term. **The matching lower bound in Theorem 2 establishes optimality up to logarithmic factors.** For GLBM, we further obtain $\widetilde{O}\big(d(M+1)\Delta_\mu + \sigma d\sqrt{T} + d^2\kappa v\rho^2\Delta_\mu\big)$ in Section 5. Here, Δ_μ is the range of reward means, κ is the worst-case inverse slope of the link function, and v and ρ govern reward dispersion and relative slope variation, respectively. This bound retains the separation between memory and statistical costs, with an additional curvature-induced error. The lower bound in Theorem 4 shows that the first two terms are near-optimal. Detailed comparisons of theoretical results are in Table 1.
- **We evaluate our algorithms on synthetic instances and two semi-synthetic caching tasks in LLM serving.** Synthetic experiments examine performance over T and sensitivity to M, R, and S, with RSM-LinUCB reducing cumulative regret by at least

61% relative to the baselines. For KV caching, we use the Mooncake conversation trace (Qin et al., 2025) to evaluate prefix-cache refresh decisions. On this task, RSM-LinUCB reduces cumulative regret by at least 20%. For semantic caching, we use vCache/SemBenchmarkLmArena (Schroeder et al., 2025). The feedback is exponential with a reciprocal mean function, yielding a generalized linear reward model. On this task, RSM-GLMUCB reduces cumulative regret by at least 46% relative to the baselines.

Table 1: Comparison of regret bounds for the basic LBM setting and the extended GLBM setting.

Comparisons for LBM	**Algorithm**	**Regret**
Jaksch et al. (2010)	UCRL2	$\widetilde{O}\big(RSMn^{M+1/2}\sqrt{T} + \sigma n^{(M+1)/2}\sqrt{T}\big)$
Clerici et al. (2023)	OFUL-memory	$\widetilde{O}\big(R(S+\sigma)\sqrt{dMT^{3/4}}\big)$
Ann et al. (2026)	OFUL-memory (improved)	$\widetilde{O}\big(RS\sqrt{dMT} + \sigma\max\{1,S\}\sqrt{d\max\{d,R^2\}T}\big)$
Ours (Upper Bound)	RSM-LinUCB Theorem 1	$\widetilde{O}\big(dRSM + \sigma d\sqrt{T}\big)$
Ours (Lower Bound)	Theorem 2	$\widetilde{\Omega}\big(dRSM + \sigma d\sqrt{T}\big)$
Comparisons for GLBM	**Algorithm**	**Regret**
Ann et al. (2026)	GLBM-SCB	$\widetilde{O}\big(\tau_w + \Delta_\mu\sqrt{MT} + C_\mu\sigma d\sqrt{T} + \Gamma_\mu d^2\kappa v\rho^2\Delta_\mu\big)$
Ours (Upper Bound)	RSM-GLMUCB Theorem 3	$\widetilde{O}\big(dM\Delta_\mu + \sigma d\sqrt{T} + d^2\kappa v\rho^2\Delta_\mu\big)$
Ours (Lower Bound)	Theorem 4	$\widetilde{\Omega}\big(dM\Delta_\mu + \sigma d\sqrt{T}\big)$

Notes. We assume $M \geq 1$ as the models reduce to standard (generalized) linear bandits when $M = 0$. Here, $n = |\mathcal{A}|$ denotes the action set size, and τ_w denotes the cumulative warm-up regret in Ann et al. (2026), which may not be sublinear in T when insufficient feature coverage prevents the warm-up stopping condition from being met. For $\rho > 0$, C_μ and Γ_μ are parameter-dependent factors satisfying $C_\mu \geq (1+\rho RS)^{3/2}$ and $\Gamma_\mu \gtrsim d + (MT)^{1/4}$. For GLBM, σ denotes an upper bound on the conditional standard deviation.

2 Problem Setup

We consider linear bandits with memory (LBM). Let $[T] \coloneqq \{1, \ldots, T\}$, and let $\mathcal{A} \subseteq \mathbb{R}^d$ be an action set satisfying $\sup_{a\in\mathcal{A}} \|a\|_2 \leq 1$. At each round $t \in [T]$, the learner selects an action $a_t \in \mathcal{A}$ based on the observed history and receives a noisy reward y_t. The reward depends on the current action, a fixed parameter $\theta^\star \in \mathbb{R}^d$, and the memory state s_t. The memory state $s_t \coloneqq (a_{t-M}, \ldots, a_{t-1})$ records the preceding M actions, where M is the known memory length. We denote the memory state space by $\mathcal{S} \coloneqq \mathcal{A}^M$ and fix an initial memory state $s_1 = (a_{1-M}, \ldots, a_0)$. For notation simplicity, we use $f(s_t, a_t) = s_{t+1} = (a_{t-M+1}, \cdots, a_t)$ to denote the memory state transition. Specifically, the observed reward is

$$y_t = \langle a_t, A(s_t)\theta^\star\rangle + \varepsilon_t. \tag{1}$$

The reward parameter $\theta^\star$ is unknown and satisfies $\theta^\star \in \Theta \coloneqq \{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq S\}$. The memory map $A : \mathcal{S} \to \mathbb{R}^{d\times d}$ is known a priori and uniformly bounded in operator norm: $\sup_{s\in\mathcal{S}} \|A(s)\|_{\mathrm{op}} \leq R$. The noise sequence $(\varepsilon_t)_{t\geq 1}$ is conditionally zero-mean and σ-sub-Gaussian given the history and the current action. Defining the effective feature $\phi(s, a) \coloneqq A(s)^\top a$, we can express the conditional mean reward as

$$r(s, a) = \langle a, A(s)\theta^\star\rangle = \langle \phi(s, a), \theta^\star\rangle .$$

The learner aims to maximize the cumulative mean reward. We compare the learner with the optimal length-T action sequence, whose cumulative mean reward is

$$\mathrm{OPT}_T(\theta^\star) \coloneqq \sup_{\bar{a}_1,\ldots,\bar{a}_T\in\mathcal{A}} \sum_{t=1}^{T} \langle \bar{a}_t, A(\bar{a}_{t-M}, \ldots, \bar{a}_{t-1})\theta^\star\rangle . \tag{2}$$

Finally, for a policy π generating actions $a_1^\pi, \ldots, a_T^\pi$, we define its policy regret as

$$\mathrm{Reg}_T(\pi) \coloneqq \mathrm{OPT}_T(\theta^\star) - \sum_{t=1}^{T} \big\langle a_t^\pi, A(a_{t-M}^\pi, \ldots, a_{t-1}^\pi)\theta^\star\big\rangle . \tag{3}$$

Remark 1 (General memory matrix.). *Prior work on LBM (Clerici et al., 2023; Ann et al., 2026) primarily considers the structured additive memory map* $A(s_t) = \left(I + \sum_{k=1}^{M} a_{t-k} a_{t-k}^\top\right)^\gamma$ *and points out that their methods can be be generalized to a general map. In this paper, we relax this structural restriction by assuming a general map* $A : \mathcal{S} \to \mathbb{R}^{d\times d}$ *with only the boundedness condition* $\sup_{s\in\mathcal{A}^M} \|A(s)\|_{\mathrm{op}} \le R$*, thereby accommodating a broader and more general class of order-sensitive and cross-action memory dynamics.*

3 Bellman-Centric Algorithm Design

We introduce *Rare-Switch Memory LinUCB* (RSM-LinUCB; see Algorithm 1). This algorithm is motivated by the observation that LBM lies at the interface between stochastic linear bandits and reinforcement learning (RL). The most recent M actions in LBM naturally form a state with known, deterministic transitions, while rewards are governed by a shared unknown parameter, as in linear bandits. Thus, LBM requires non-myopic planning, as in RL, but only the reward parameter needs to be learned from bandit feedback.

From the bandit side, we construct an optimistic reward function $u_k(s, a)$ using a linear bandit confidence set. From the RL side, we solve the corresponding Bellman equation to obtain a value function $J_k(s)$ and select actions by accounting for both immediate optimism and the value of the resulting memory state. This non-myopic planning step goes beyond the batching-based treatment of memory used in prior LBM algorithms (Clerici et al., 2023; Ann et al., 2026). To prevent frequent policy changes from repeatedly inducing memory-state transients, we adopt a determinant-triggered rare-switching rule (Abbasi-Yadkori et al., 2011) and replan only after the accumulated statistical information has increased substantially.

Algorithm 1 Rare-Switch Memory LinUCB

Input: Confidence level $\delta \in (0, 1)$, regularization $\lambda > 0$, discount factor γ
Initialize: $V_0 \leftarrow \lambda I_d,\ b_0 \leftarrow 0_d,\ t \leftarrow 1,\ k \leftarrow 0$
while $t \le T$ **do**
  $k \leftarrow k + 1,\ \tau_k \leftarrow t,\ \overline{V}_k \leftarrow V_{t-1},\ \overline{b}_k \leftarrow b_{t-1},\ \overline{\theta}_k \leftarrow V_{t-1}^{-1} b_{t-1}$
  while $t \le T$ **and** $\det(V_{t-1}) < 2\det(\overline{V}_k)$ **do**
    Play a_t according to Equation (4) with $u_k(s, a)$ and $J_k(s)$ defined as in Equations (5) and (6), and observe reward y_t
    $V_t \leftarrow V_{t-1} + \phi(s_t, a_t)\phi(s_t, a_t)^\top,\ b_t \leftarrow b_{t-1} + y_t\phi(s_t, a_t),\ s_{t+1} \leftarrow f(s_t, a_t),\ t \leftarrow t + 1$
  end while
end while

Input and initialization. The algorithm takes a confidence level δ, a regularization parameter λ, and a discount factor γ as inputs. It initializes the information matrix and reward feature vector as $V_0 = \lambda I_d$ and $b_0 = 0_d$, respectively, and sets the episode counter $k = 0$. The algorithm proceeds in episodes, during each of which the reward estimator and the resulting policy remain fixed to limit the frequency of policy changes and to control the cost of the memory state transients.

Action selection with frozen parameters. At the beginning of episode k, the algorithm records the starting round τ_k and freezes the current information matrix $\overline{V}_k$, reward feature vector $\overline{b}_k$, and parameter estimate $\overline{\theta}_k$. At round t within episode k, the algorithm selects an action that maximizes the optimistic action value, given by

$$a_t \in \arg\max_{a\in\mathcal{A}} \left\{u_k(s_t, a) + \gamma J_k(f(s_t, a))\right\}, \tag{4}$$

where $u_k(s, a)$ denotes the optimistic reward estimate and $J_k(s)$ the corresponding optimal discounted value, defined as

$$u_k(s, a) = \Pi_{[-RS,RS]}\left(\phi(s, a)^\top \overline{\theta}_k + \overline{\beta}_k \|\phi(s, a)\|_{\overline{V}_k^{-1}}\right), \tag{5}$$

$$J_k(s) = \sup_{\overline{a}_0,\overline{a}_1,\ldots} \sum_{j=0}^{\infty} \gamma^j u_k(\overline{s}_j, \overline{a}_j). \tag{6}$$

Here, $\overline{\beta}_k = \sqrt{\lambda}S + \sigma\sqrt{2\log\left(\frac{1}{\delta}\right) + d\log\left(1 + \frac{\tau_k R^2}{\lambda d}\right)}$ is the confidence level, $\Pi_{[-RS,RS]}$ projects the estimate to the range of rewards, and $\overline{s}_0 = s$, $\overline{s}_{j+1} = f(\overline{s}_j, \overline{a}_j)$ is the memory-state trajectory. With high probability, $u_k(s,a)$ upper bounds the expected reward of taking action a in memory state s. Built on this optimistic reward function, $J_k(s)$ accounts for the optimistic cumulative reward obtainable from state s by recursively evaluating future action sequences and their induced memory-state transitions. The algorithm plays a_t and observes the reward y_t.

Determinant-triggered policy updates. The agent updates V_t, b_t, and the memory state s_{t+1} using the selected action and the observed reward. The algorithm uses a determinant-triggered rare-switching rule, starting a new episode only when $\det(V_t) \geq 2\det(\overline{V}_k)$. Thus, the estimator used for planning and the resulting policy remain fixed within each episode, even as new observations continue to accumulate. The selected actions may nevertheless vary as the memory state evolves. This update rule limits the number of policy changes and the memory state transients. Therefore, any two initial memories at the start of episode become identical after the same sequence of M actions, allowing us to bound the transient contribution through an M-dependent value span. Unlike the predetermined batching policies used in prior work on LBM (Clerici et al., 2023; Ann et al., 2026), our episode boundaries adapt to the information accumulated by the learner. The determinant-doubling rule therefore controls the frequency of replanning while ensuring that each policy update is triggered by a substantial increase in statistical information.

Remark 2 (Computation of J_k)**.** *Action selection in Equation (4) requires the value function J_k, which is fixed per episode; within the episode, each round reduces to a maximization over a single action. For small-scale problems where the memory state space $\mathcal{S}$ is not large, J_k can be computed near-exactly by discounted Howard policy iteration, and we show in Appendix C.1 that the resulting approximation error is negligible, so the regret order is preserved as in Section 4. For large-scale problems, the planning step may use numerical optimization methods such as gradient ascent, which have also been used for optimistic arm selection in LBM (see, e.g., Clerici et al., 2023; Ann et al., 2026). Moreover, application structure often admits more efficient planners: for example, in our cache experiments in Section 6, the memory state is determined by the set of refreshed items, so planning reduces to a combinatorial optimization that can be solved exactly by tree dynamic programming or sorting (see Appendix C for the details).*

4 Theoretical Guarantee

4.1 Upper Bound of RSM-LinUCB

Our analysis first establishes that, under any bounded reward function, the optimal discounted values of two memory states differ by at most $2RSM$.

Lemma 1 (Finite-memory value span)**.** *Let $\gamma \in (0,1)$ and $r_q : \mathcal{S} \times \mathcal{A} \to [-RS, RS]$ be any reward function. Define $J_q(s) = \sup_{a_0,a_1,\ldots} \sum_{j=0}^{\infty} \gamma^j r_q(s_j, a_j)$, where $s_0 = s$ and $s_{j+1} = f(s_j, a_j)$. Then*

$$\operatorname{span}(J_q) \coloneqq \sup_{s\in\mathcal{S}} J_q(s) - \inf_{s\in\mathcal{S}} J_q(s) \leq 2RSM.$$

The key insight behind Lemma 1 is that applying the same sequence of M actions from any two initial memory states makes the resulting states coincide. The initial history at each episode can therefore affect rewards during the first M rounds, which bounds the effect of the starting state at each episode boundary. Combining this property with the Bellman optimality equation and optimism yields the following regret decomposition.

Lemma 2 (Bellman-centric regret decomposition)**.** *Consider Algorithm 1 with $T \geq 2$ and $\gamma = 1 - 1/T$. Let π denote the policy generated by Algorithm 1, K the total number of episodes, and $k(t)$ the episode containing round t. Then, with probability at least $1-\delta$,*

$$\operatorname{Reg}_T(\pi) \leq 2(K+2)RSM + \sum_{t=1}^{T} \left[u_{k(t)}(s_t^\pi, a_t^\pi) - r_{\theta^\star}(s_t^\pi, a_t^\pi)\right].$$

Lemma 2 separates the regret into a *memory-induced error* and the cumulative reward *estimation error* along the learner's trajectory. The first term bounds the error of starting each episode from

a potentially unfavorable memory state. Our determinant-doubling rule ensures $K = O(d \log T)$, so this error contributes only an additive memory-dependent term to the regret bound. The second term can be controlled by using ridge-regression confidence bounds and the elliptical potential lemma (Abbasi-Yadkori et al., 2011). Specifically, the determinant doubling trick ensures that freezing the estimator within each episode increases the confidence intervals by at most a constant factor.

Theorem 1 (Upper Bound of Algorithm 1). *For $T \geq 2$, with inputs $0 < \lambda \leq d\sigma^2/S^2$, and $\gamma = 1 - 1/T$, the regret of Algorithm 1 satisfies*

$$\begin{aligned}\mathrm{Reg}_T(\text{RSM-LinUCB}) &\leq \left(4d \log\left(1 + \frac{TR^2}{\lambda d}\right) + 6\right) RSM + 4RSd \log\left(1 + \frac{TR^2}{\lambda d}\right) \\ &\quad + 4\sqrt{Td \log\left(1 + \frac{TR^2}{\lambda d}\right)} \left(\sigma \sqrt{d \log\left(1 + \frac{TR^2}{\lambda d}\right) + 2\log\left(\frac{1}{\delta}\right)} + \sqrt{\lambda} S\right) \\ &= \widetilde{O}\big(dRS(M+1) + \sigma d\sqrt{T}\big)\end{aligned}$$

with probability at least $1 - \delta$, where $\widetilde{O}$ omits logarithmic factors.

The regret bound consists of two additive terms. The first, $\widetilde{O}(dRS(M+1))$, captures the memory-induced error of starting each episode from a potentially unfavorable state. The finite memory value-span bound limits this error to $O(RSM)$ per episode, while determinant doubling limits the number of episodes to $\widetilde{O}(d)$. The additional $\widetilde{O}(dRS)$ term accounts for the initial exploration error present even without memory that also exists in ordinary linear bandits. The second term, $\widetilde{O}(\sigma d\sqrt{T})$, accounts for reward estimation error. Under the prescribed regularization, this term has no polynomial dependence on the parameter-scale factors R (for memory matrix A) or S (for reward parameter θ^*).

Remark 3. *Compared with the best upper bound $\widetilde{O}\big(RS\sqrt{dMT} + \sigma \max\{1, S\}\sqrt{d \max\{d, R^2\}T}\big)$ from prior work (Ann et al., 2026), our result not only clearly separates the memory-dependent cost from the $\sqrt{T}$ estimation error term, but also removes the additional parameter-scale factors (R and S) from the latter (see Table 1 for a detailed comparison). Thus, larger M or R only increases the additive memory term rather than the coefficient of the leading $\sqrt{T}$ term, up to logarithmic factors. This separation is particularly beneficial when the memory length M is large. The lower bound below establishes that these dependencies are near-optimal up to logarithmic factors.*

4.2 Lower Bound

Theorem 2 (Lower Bound of LBM). *Suppose $d \geq 2$ and $T \geq d \max\left\{M, 1, \sigma^2 d / \left(48R^2S^2\right)\right\}$. There exists an admissible action set such that the minimax regret of LBM satisfies*

$$\inf_{\pi} \sup_{\substack{\theta^\star:\, \|\theta^\star\|_2 \leq S \\ A:\, \|A\|_{\mathrm{op}} \leq R}} \mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \geq \Omega\big(dRS(M+1) + \sigma d\sqrt{T}\big).$$

The upper bound in Theorem 1 matches the lower bound in Theorem 2 up to logarithmic factors. The first memory-induced error reflects the cost of obtaining information that is accessible only through a suitable action history. In our hard instance, the learner must repeatedly select a coordinate direction for M rounds before rewards can reveal whether that coordinate is informative. Identifying the rewarding coordinate among d candidates can therefore require $\Omega(dM)$ rounds, with a reward gap of order RS during unsuccessful exploration. This yields the $\Omega(dRSM)$ term, while the additional $\Omega(dRS)$ term accounts for the initial exploration error already present in ordinary linear bandits, giving $\Omega(dRS(M+1))$ overall. The statistical estimation error $\Omega(\sigma d\sqrt{T})$ follows from the standard lower bound in linear bandits (Lattimore & Szepesvári, 2020), since ordinary linear bandits are a special case of LBM with a constant memory matrix.

5 Extension to Generalized Linear Bandits with Memory

Our Bellman-centric design achieves near-optimal regret for LBM. We next ask whether the same idea can improve prior bounds for generalized linear rewards (Ann et al., 2026) and yield near-optimal leading regret terms, as summarized in Table 1. The extension is not immediate: in generalized linear models the slope of the link function depends on the unknown parameter, which makes

uncertainty estimation considerably harder than in the linear case. In this section, we extend our algorithm to generalized rewards, show that it achieves an additive memory error, and prove that the leading terms of the resulting bound are near-optimal.

5.1 Problem Setup

Generalized linear bandits with memory (GLBM) retain the same action space $\mathcal{A}$, known memory matrix $A(\cdot)$, and norm bounds R and S as in LBM. Similarly, we write $\phi(s,a) = A(s)^\top a$ and $z(s,a) = \phi(s,a)^\top \theta^*$. Unlike LBM, where the conditional reward mean is $z(s,a)$, GLBM models the reward through a known canonical exponential family (Lee et al., 2024; Liu et al., 2024; Ann et al., 2026). Specifically, an observation y at (s,a) has density or probability mass function

$$\mathbb{P}[y \mid z(s,a)] = \exp\left(\frac{yz(s,a) - b(z(s,a))}{v} + c(y,v)\right),$$

where $v > 0$ is the known dispersion parameter which controls the noise level of the observed reward, and $\mu(\cdot) = b'(\cdot)$ is the known link function. The conditional reward mean and variance are therefore given by $\mathbb{E}[y \mid z(s,a)] = \mu(z(s,a))$ and $\mathrm{Var}[y \mid z(s,a)] = v\mu'(z(s,a))$, respectively.

Following prior work, we assume that μ is twice continuously differentiable and satisfies $\mu'(z) > 0$ and $|\mu''(z)| \le \rho\mu'(z)$ for every $|z| \le RS$, where the known constant $\rho \ge 0$ controls the relative variation of its slope. We further define

$$U_\mu \coloneqq \sup_{|z|\le RS} \mu'(z),\ \kappa \coloneqq \Big(\inf_{|z|\le RS} \mu'(z)\Big)^{-1},\ \Delta_\mu \coloneqq \mu(RS) - \mu(-RS). \tag{7}$$

Here, U_μ bounds how fast the reward changes as $z(s,a)$ varies, κ quantifies the worst-case flatness of the link function, and Δ_μ bounds the range of reward means. The definitions of OPT_T and Reg_T remain unchanged from Section 2, with the mean reward replaced by $r(s,a) = \mu(\phi(s,a)^\top \theta^\star)$.

5.2 Algorithm and Theoretical Guarantee

In GLBM, the slope of the mean reward function depends on the unknown parameter, making uncertainty estimation more challenging than in the linear model. We therefore propose *Rare-Switch Memory GLMUCB* (RSM-GLMUCB; see Algorithm 2), which uses maximum likelihood estimation (MLE) to construct confidence sets $\mathcal{C}_k$ at each episode while retaining the discounted planning approach of Algorithm 1. The algorithm intersects each new confidence set with the previous one so that uncertainty does not increase across episodes. We also maintain two information matrices V_t and H_t: the former one uses all selected actions, while the latter one weights them by the estimated reward slope when predictions are sufficiently accurate. These weights, similar to the rare-switch policy in Algorithm 1, are fixed once recorded, and we update the policy only when either determinant doubles. This allows the algorithm to account for generalized rewards while keeping policy updates infrequent. Due to space limits, we defer further details of Algorithm 2 to Appendix B.

Algorithm 2 Rare-Switch Memory GLMUCB

Input: Confidence level $\delta \in (0,1)$; regularization $\lambda_1, \lambda_2 > 0$; discount factor $\gamma \in (0,1)$.
Initialize: $V_0 \leftarrow \lambda_1 I_d$, $H_0 \leftarrow \lambda_2 I_d$, $\mathcal{C}_0 \leftarrow \Theta$, $t \leftarrow 1$, $k \leftarrow 0$.
while $t \le T$ **do**
  $k \leftarrow k+1$, $\tau_k \leftarrow t$, $\overline{V}_k \leftarrow V_{t-1}$, $\overline{H}_k \leftarrow H_{t-1}$.
  Compute $\widehat{\theta}_k$, $\mathcal{C}_k$, and $\overline{\theta}_k$ according to Equation (9).
  while $t \le T$ and $\max\left\{\frac{\det(V_{t-1})}{\det(\overline{V}_k)}, \frac{\det(H_{t-1})}{\det(\overline{H}_k)}\right\} < 2$ **do**
    Play a_t according to Equation (12) with $u_k(s,a)$ and $J_k(s)$ defined as in Equation (13), and observe y_t. Then compute w_t according to Equation (14).
    $V_t = V_{t-1} + \phi(s_t,a_t)\phi(s_t,a_t)^\top$, $H_t = H_{t-1} + w_t\phi(s_t,a_t)\phi(s_t,a_t)^\top$, $s_{t+1} \leftarrow f(s_t,a_t)$, $t \leftarrow t+1$.
  end while
end while

Theorem 3 (Upper bound of Algorithm 2). *For $T \geq 2$, $0 < \lambda_1 \leq \kappa v d/S^2, 0 < \lambda_2 \leq v d/S^2, \gamma = 1 - 1/T$. Under generalized reward, with probability at least $1-\delta$, the regret of Algorithm 2 satisfies*

$$\mathrm{Reg}_T(\text{RSM-GLMUCB}) \leq \widetilde{O}\big(d(M+1)\Delta_\mu + d\sqrt{vU_\mu T} + d^2\kappa v\rho^2\Delta_\mu\big).$$

The first is the memory-induced error: over $\widetilde{O}(d)$ episodes, the M-dependent transient cost is at most $M\Delta_\mu$ per episode, with an initial exploration cost of $\widetilde{O}(d\Delta_\mu)$. The second term accounts for parameter estimation error, where $\sqrt{vU_\mu}$ bounds the conditional standard deviation of the observation noise, denoted by σ in Table 1. These two terms have the similar interpretation as their counterparts in Theorem 1. The third term captures the additional cost of learning under parameter-dependent curvature. Here, κ reflects how small the derivative of the link function can be, while ρ controls its relative variation as the predicted reward parameter changes. Our analysis bounds the number of rounds with large prediction uncertainty or insufficient information by $\widetilde{O}(d^2\kappa v\rho^2 + d)$, and charges at most Δ_μ estimation error to each such round. When $\rho = 0$, the link function is affine and the curvature dependence vanishes, and this bound reduces to Theorem 1 under linear reward.

Theorem 4 (Lower bound of GLBM). *Let $d \geq 3$, $M \geq 0$, and $T \geq \max\big\{dM, d, \frac{v(d-1)^2}{U_\mu}\max\big\{\frac{1}{4R^2S^2}, \rho^2\big\}\big\}$. Then there exists an admissible action set such that*

$$\inf_{\pi}\sup_{\theta^\star, A}\mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \geq \widetilde{\Omega}\Big(d(M+1)\Delta_\mu + d\sqrt{vU_\mu T}\Big). \tag{8}$$

This lower bound matches the first two terms of the upper bound in Theorem 3 up to logarithmic factors. Thus, both the memory-induced error and the estimation error are unavoidable in GLBM, consistent with our findings for LBM. The remaining gap is the curvature-induced identification cost: our upper bound controls it through worst-case curvature, while the lower bound does not establish whether this dependence is necessary. Closing this gap remains an interesting direction. Compared with (Ann et al., 2026), our upper bound offers three improvements. It avoids a separate warm-up phase whose regret can be linear in T; isolates the dependence on M in an additive memory term while retaining the optimal $\sqrt{T}$ estimation rate; and removes the additional M- and T-dependence from the curvature-induced error while improving its dimension dependence (see Table 1).

6 Numerical Experiments

We evaluate our algorithms on synthetic data and semi-synthetic datasets for KV- and semantic caching. For the LBM experiments, including the synthetic and KV-cache experiments, we compare RSM-LinUCB (ours) with OFUL (original) (Clerici et al., 2023), its refined-block variant (OFUL (refined)), GLBM-SCB adapted to linear rewards (Ann et al., 2026), and Myopic LinUCB (Abbasi-Yadkori et al., 2011). For the GLBM experiment on semantic caching, we compare RSM-GLMUCB (ours) with GLBM-SCB and the per-round GLB-OMD algorithm (Zhang et al., 2026). We measure cumulative pseudo-regret against the exact finite horizon optimum under the same initial memory. More implementation details and parameter choices are provided in Appendix C.

Synthetic data experiments. We consider LBM instances with $d = 10$, $\mathcal{A} = \{e_1, \ldots, e_{10}\}$, and independent Gaussian noise of standard deviation $\sigma = 0.25$. The memory matrix is $Re_ie_i^\top$ if the preceding M actions are all e_i, and zero otherwise, so switching actions delays informative rewards. We set $\theta^\star = Su$ for a fixed unit direction u and use $\gamma = 0.9999$. We record a continuous trajectory up to $T = 10^6$ and separately vary $M \in \{100, 200, \ldots, 1000\}$ and $R, S \in \{10, 20, \ldots, 100\}$, one parameter at a time. Results average 30 noise seeds, with ten balanced initial histories averaged within each seed, and are presented in Figure 1 (A) - (D). After initial exploration, RSM-LinUCB achieves the lowest regret among all algorithms across the tested values of M, R, and S.

KV cache. We use Mooncake conversation dataset (Qin et al., 2025), containing 12,031 LLM requests whose anonymized hashes identify shared 512-token prefix blocks. Each action refreshes one candidate prefix, whose KV blocks remain available for the current and following M rounds. The reward is the normalized average number of reused prefix tokens across 36 held-out requests, a proxy for avoided prefill computation; its conditional expectation is linear in the post-refresh cache-availability feature, yielding an LBM instance. We simulate refresh decisions under a 3×10^6-token

cache budget, where each round represents 10 seconds and $M = 180$ corresponds approximately to Mooncake Store's default 30-minute soft-pin lifetime (Mooncake Team, 2026). To reduce communication overhead and limit exposure of individual access patterns (LMCache; Wu et al., 2025), we consider a lightweight refresh controller that receives aggregate cache-reuse statistics rather than request-level prefix logs. We set $T = 10^5$, $M = 180$, and choose $M \in \{30, 60, 90, 120, 150, 180\}$, using paired request streams and three seeds, see Figure 1 (E) - (F). RSM-LinUCB achieves lower regret in the time comparison and the lowest mean regret throughout the memory sweep. We also note that at large M, regret increases slower, which is consistent with the fact that longer retention keeps more useful prefixes resident and reduces losses from missed reuse.

Semantic cache. We use `vCache/SemBenchmarkLmArena` (Schroeder et al., 2025), which contains real LMArena queries, generated paraphrases, responses, embeddings, and semantic labels. Unlike exact prefix reuse in KV caching, semantic caching reuses complete answers and may return an unsuitable response. Effective semantic caching must therefore reduce model calls while maintaining reuse correctness (Bang, 2023; Schroeder et al., 2025). We hence measure how long the cache sustains correct, cache-only service before the first model fallback or incorrect reuse. Each round, the controller refreshes one bundle without observing incoming queries, which then remains available for the current and following M rounds. We simulate unit-rate Poisson arrivals with independently sampled queries and use benchmark labels to assess correctness. For each simulated waiting-time measurement, we hold the resulting cache configuration fixed. If $z = \phi(s, a)^\top \theta^\star \in [0, 1)$ denotes the rate of requests correctly served by the cache, this duration is exponentially distributed with mean $\mu(z) = 1/(1 - z)$, yielding a generalized linear reward model. With a fixed 64 MiB payload budget, we report a $T = 10^6$ run at $M = 51$ and a sweep over $M \in \{10, 20, 24, 30, 40, 50\}$, using three seeds, see Figure 1 (G) - (H). RSM-GLMUCB outperforms both baselines and achieves lower mean regret at every tested value of M. At larger M, its regret grows more slowly, consistent with longer retention improving cache coverage.

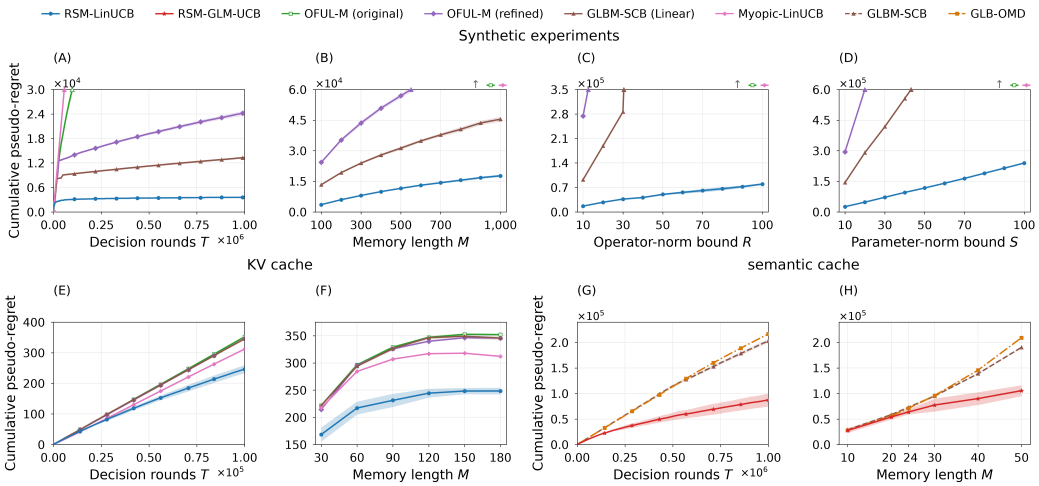

Figure 1: Cumulative pseudo-regret for synthetic (A–D), KV-cache (E–F), and semantic-cache (G–H). Shading denotes one standard deviation and some baseline curves exceed the displayed range.

7 Conclusion

We developed RSM-LinUCB and RSM-GLMUCB for LBM and GLBM, combining linear bandit reward estimation and determinant-triggered rare switching with Bellman planning over memory states. Our analysis separates memory-induced error from estimation error, with leading regret terms matching the lower bounds up to logarithmic factors. Future work includes closing the gap in the curvature-induced term for GLBM by sharpening the upper bound or proving a matching lower bound. Another direction is to explore whether our action-dependent memory framework can help develop theoretical models of memory management in LLM-based systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with formalizing proofs, implementing methods, and cleaning and reformatting datasets. We did not use generative AI tools for the remaining required-disclosure tasks, including generating synthetic datasets, helping develop theoretical models or conceptual frameworks, designing or providing feedback on research methodology or experiments, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, assisting with translation, supporting qualitative or thematic data analysis, or interpreting results. Additionally, we used generative AI tools for editing to improve readability. We reviewed all AI-assisted work. We checked AI-assisted proofs step by step for mathematical correctness and consistency with the stated results. We reviewed the experimental implementations against the intended methods and tested the code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Neural Information Processing Systems*, 2011.

Marc Abeille and Alessandro Lazaric. Linear Thompson Sampling Revisited. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 176–184. PMLR, 20–22 Apr 2017.

Marc Abeille, Louis Faury, and Clement Calauzenes. Instance-wise minimax-optimal algorithms for logistic bandits. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3691–3699. PMLR, 13–15 Apr 2021.

Priyank Agrawal and Theja Tulabandula. Learning by repetition: Stochastic multi-armed bandits under priming effect. In *Conference on Uncertainty in Artificial Intelligence*, pp. 470–479. PMLR, 2020.

Heesang Ann, Hyunjun Choi, Taehyun Hwang, Younghoon Shin, Haeju Cheong, and Min-hwan Oh. Generalized linear bandits with memory. In *Forty-third International Conference on Machine Learning*, 2026.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 138–158. PMLR, 25–28 Jun 2019.

Pranjal Awasthi, Kush Bhatia, Sreenivas Gollapudi, and Kostas Kollias. Congested bandits: Optimal routing via short-term resets. In *International Conference on Machine Learning*, pp. 1078–1100. PMLR, 2022.

Fu Bang. GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pp. 212–218, Singapore, December 2023. Association for Computational Linguistics.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, 27, 2014.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free. In *Conference on Learning Theory*, pp. 696–726. PMLR, 2019.

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.

Giulia Clerici, Pierre Laforgue, and Nicolo Cesa-Bianchi. Linear bandits with memory: from rotting to rising. *arXiv preprint arXiv:2302.08345*, 2023.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory*, number 101, pp. 355–366, 2008.

Arnoud V den Boer and N Bora Keskin. Dynamic pricing with demand learning and reference effects. *Management Science*, 68(10):7112–7130, 2022.

Giuseppe Di Benedetto, Vito Bellini, and Giovanni Zappella. A linear bandit for seasonal environments. *arXiv preprint arXiv:2004.13576*, 2020.

Louis Faury, Marc Abeille, Clement Calauzenes, and Olivier Fercoq. Improved optimistic algorithms for logistic bandits. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3052–3060. PMLR, 13–18 Jul 2020.

Louis Faury, Marc Abeille, Kwang-Sung Jun, and Clement Calauzenes. Jointly efficient and optimal algorithms for logistic bandits. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 546–580. PMLR, 28–30 Mar 2022.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. *Advances in neural information processing systems*, 23, 2010.

Ayoub Foussoul, Vineet Goyal, Orestis Papadigenopoulos, and Assaf Zeevi. Last switch dependent bandits with monotone payoff functions. In *International Conference on Machine Learning*, pp. 10265–10284. PMLR, 2023.

Yao Fu, Leyang Xue, Yeqi Huang, Andrei-Octavian Brabete, Dmitrii Ustiugov, Yuvraj Patel, and Luo Mai. ServerlessLLM: Low-Latency serverless inference for large language models. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 135–153, 2024.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *International conference on algorithmic learning theory*, pp. 174–188. Springer, 2011.

Saeed Ghoorchian, Evgenii Kortukov, and Setareh Maghsudi. Non-stationary linear bandits with dimensionality reduction for large-scale recommender systems. *IEEE Open Journal of Signal Processing*, 5:548–558, 2024. doi: 10.1109/OJSP.2024.3386490.

Botao Hao, Tor Lattimore, and Csaba Szepesvari. Adaptive exploration in linear contextual bandit. In *International Conference on Artificial Intelligence and Statistics*, pp. 3536–3545. PMLR, 2020.

Kihyuk Hong, Yuhang Li, and Ambuj Tewari. An optimization-based algorithm for non-stationary kernel bandits without prior knowledge. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 3048–3085. PMLR, 25–27 Apr 2023.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

Huiwen Jia, Cong Shi, and Siqian Shen. Online learning and pricing for network revenue management with reusable resources. *Advances in Neural Information Processing Systems*, 35:4830–4842, 2022a.

Huiwen Jia, Cong Shi, and Siqian Shen. Online learning and pricing with reusable resources: Linear bandits with sub-exponential rewards. In *International Conference on Machine Learning*, pp. 10135–10160. PMLR, 2022b.

Huiwen Jia, Cong Shi, and Siqian Shen. Online learning and pricing for service systems with reusable resources. *Operations Research*, 72(3):1203–1241, 2024.

Kwang-Sung Jun, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. *Advances in Neural Information Processing Systems*, 30, 2017.

Zohar S Karnin and Oren Anava. Multi-armed bandits: Competing with optimal sequences. *Advances in Neural Information Processing Systems*, 29, 2016.

Khashayar Khosravi, Renato Paes Leme, Chara Podimata, and Apostolis Tsorvantzis. Preferences evolve and so should your bandits: Bandits with evolving states for online platforms. *arXiv preprint arXiv:2307.11655*, 2023.

Baekjin Kim and Ambuj Tewari. Randomized exploration for non-stationary stochastic linear bandits. In *Conference on Uncertainty in Artificial Intelligence*, pp. 71–80. PMLR, 2020.

Pierre Laforgue, Giulia Clerici, Nicolò Cesa-Bianchi, and Ran Gilad-Bachrach. A last switch dependent analysis of satiation and seasonality in bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 971–990. PMLR, 2022.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Cheung Hao Lee and Patrick Wong. Drift-aware llm routing with sparse contexts and shared budgets. *arXiv preprint arXiv:2609.00662*, 2026.

Junghyun Lee, Se-Young Yun, and Kwang-Sung Jun. A unified confidence sequence for generalized linear models, with applications to bandits. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024.

Liu Leqi, Fatma Kilinc Karzan, Zachary Lipton, and Alan Montgomery. Rebounding bandits for modeling satiation effects. *Advances in Neural Information Processing Systems*, 34:4003–4014, 2021.

Chuanhao Li, Qingyun Wu, and Hongning Wang. Unifying clustered and non-stationary bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 1063–1071. PMLR, 2021.

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2071–2080. PMLR, 06–11 Aug 2017.

Shuai Liu, Alex Ayoub, Flore Sentenac, Xiaoqi Tan, and Csaba Szepesvári. Almost free: Self-concordance in natural exponential families and an application to bandits. *Advances in Neural Information Processing Systems*, 37:70828–70874, 2024.

LMCache. LMCache documentation: Metrics reference. `https://docs.lmcache.ai/production/observability/metrics.html`. Accessed September 18, 2026.

Haipeng Luo, Chen-Yu Wei, Alekh Agarwal, and John Langford. Efficient contextual bandits in non-stationary worlds. In *Conference On Learning Theory*, pp. 1739–1776. PMLR, 2018.

Dhruv Malik, Yuanzhi Li, and Aarti Singh. Complete policy regret bounds for tallying bandits. In *Conference on Learning Theory*, pp. 5146–5174. PMLR, 2022.

Dhruv Malik, Conor Igoe, Yuanzhi Li, and Aarti Singh. Weighted tallying bandits: overcoming intractability via repeated exposure optimality. In *International Conference on Machine Learning*, pp. 23590–23609. PMLR, 2023.

Mooncake Team. Mooncake Store: Design and configuration documentation. `https://kvcache-ai.github.io/Mooncake/design/mooncake-store.html`, 2026. Accessed: August 14, 2026.

Ruoyu Qin, Zheming Li, Weiran He, Jialei Cui, Feng Ren, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. Mooncake: Trading more storage for less computation — a KVCache-centric architecture for serving LLM chatbot. In *23rd USENIX Conference on File and Storage Technologies (FAST 25)*, pp. 155–170, Santa Clara, CA, February 2025. USENIX Association. ISBN 978-1-939133-45-8. URL https://www.usenix.org/conference/fast25/presentation/qin.

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

Yoan Russac, Olivier Cappé, and Aurélien Garivier. Algorithms for non-stationary generalized linear bandits. *arXiv preprint arXiv:2003.10113*, 2020.

Ayush Sawarni, Nirjhar Das, Siddharth Barman, and Gaurav Sinha. Generalized linear bandits with limited adaptivity. *Advances in Neural Information Processing Systems*, 37:8329–8369, 2024.

Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. *Advances in Neural Information Processing Systems*, 26, 2013.

Luis Gaspar Schroeder, Aditya Desai, Alejandro Cuadron, Kyle Chu, Shu Liu, Mark Zhao, Stephan Krusche, Alfons Kemper, Matei Zaharia, and Joseph E Gonzalez. vcache: Verified semantic prompt caching. *arXiv preprint arXiv:2502.03771*, 2025.

Annette Taberner-Miller. Paretobandit: Budget-paced adaptive routing for non-stationary llm serving. *arXiv preprint arXiv:2604.00136*, 2026.

Wei Tang, Chien-Ju Ho, and Yang Liu. Bandit learning with delayed impact of actions. *Advances in Neural Information Processing Systems*, 34:26804–26817, 2021.

vLLM Contributors. vLLM documentation: Metrics. https://docs.vllm.ai/en/v0.12.0/design/metrics/. Version 0.12.0. Accessed September 15, 2026.

Jing Wang, Peng Zhao, and Zhi-Hua Zhou. Revisiting weighted strategy for non-stationary parametric bandits. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7913–7942. PMLR, 25–27 Apr 2023.

Romain Warlop, Alessandro Lazaric, and Jérémie Mary. Fighting boredom in recommender systems with linear reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Guanlong Wu, Zheng Zhang, Yao Zhang, Weili Wang, Jianyu Niu, Ye Wu, and Yinqian Zhang. I know what you asked: Prompt leakage via kv-cache sharing in multi-tenant llm serving. In *NDSS*, 2025.

Qingyun Wu, Naveen Iyer, and Hongning Wang. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 495–504, 2018.

Xiao Xu, Fang Dong, Yanghua Li, Shaojian He, and Xin Li. Contextual-bandit based personalized recommendation with time-varying user interests. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6518–6525, 2020.

Raymond Zhang, Hédi Hadiji, and Richard Combes. Linear bandits on ellipsoids: Minimax optimal algorithms. *arXiv preprint arXiv:2502.17175*, 2025.

Yu-Jie Zhang, Sheng-An Xu, Peng Zhao, and Masashi Sugiyama. Generalized linear bandits: Almost optimal regret with one-pass update. *Advances in Neural Information Processing Systems*, 38:69244–69277, 2026.

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.

Zifeng Zhao, Feiyu Jiang, Yi Yu, and Xi Chen. High-dimensional dynamic pricing under nonstationarity: Learning and earning with change-point detection. *Management Science*, 2026.

A Related works

Linear bandits with memory. The works most closely related to ours are the linear bandits with memory models studied by Clerici et al. (2023) and Ann et al. (2026). Clerici et al. (2023) introduce a linear bandit model in which the learner's most recent actions endogenously affect the reward parameter. Their OFUL-memory algorithm reduces the original problem to learning a cyclic policy: at the start of each episode, the learner commits to a sequence of actions, with the first M actions clearing the memory inherited from the previous episode, and updates its parameter estimate only after executing the entire sequence. This approach achieves a $\widetilde{O}(T^{3/4})$ regret bound against the optimal action sequence. Building on this algorithmic idea, Ann et al. (2026) develop a sharper episode-level analysis of OFUL-memory, yielding a regret bound of $\widetilde{O}(RS\sqrt{dMT} + \sigma \max\{1, S\}\sqrt{d \max\{d, R^2\}T})$. However, this approach still selects an entire action sequence at the start of each predetermined episode with fixed length, leaving the memory length and horizon coupled in the regret bound. In contrast, our key observation is that the most recent M actions naturally form a state with known and deterministic transitions, while rewards are governed by a shared unknown parameter $\theta^\star$, as in linear bandits. Building on this structure, RSM-LinUCB combines optimistic reward estimation with Bellman planning and determinant-triggered rare switching. For $M \geq 1$ and a general known memory map satisfying $\|A(\cdot)\|_{\mathrm{op}} \leq R$, our algorithm achieves $\widetilde{O}(dRSM + \sigma d\sqrt{T})$ regret. This separates the memory-induced error from the statistical estimation error, making the memory cost additive and independent of the horizon up to logarithmic factors. The bound matches our lower bound up to logarithmic factors. Compared with the refined OFUL-memory bound in prior work, our $dRSM$ term improves on its $RS\sqrt{dMT}$ memory term, while our $\sigma d\sqrt{T}$ estimation term removes the additional polynomial dependence on R and S. A detailed comparison is provided in Table 1.

Non-stationary bandits. Our work is related to the broader literature on non-stationary bandits. Classical non-stationary bandit models characterize environmental changes through a bounded number of breakpoints (Garivier & Moulines, 2011; Auer et al., 2019), with algorithms based on discounting, sliding windows, restarts, or adaptive change detection. Other formulations control non-stationarity through a variation budget or related regularity conditions (Besbes et al., 2014; Karnin & Anava, 2016). These ideas have subsequently been extended to linear and contextual bandits (Wu et al., 2018; Chen et al., 2019; Di Benedetto et al., 2020; Xu et al., 2020; Li et al., 2021; Luo et al., 2018; Cheung et al., 2019; Russac et al., 2019; Kim & Tewari, 2020; Hong et al., 2023), as well as to generalized linear reward models (Russac et al., 2020; Zhao et al., 2020; Wang et al., 2023). In these models, the time-varying parameter θ_t is typically generated by an exogenous process that is not controlled by the learner. By contrast, the non-stationarity in our setting is endogenous: the learner's recent actions change the memory state and thereby affect future reward functions.

In addition to LBM, several models allow past actions to influence current rewards. Recency-dependent recommendation models (Warlop et al., 2018) express each arm's reward as a polynomial of recency-weighted play counts over a finite action set. Congested bandits (Awasthi et al., 2022) model each arm's reward as depending on how often it was selected within a fixed window. Bandit-based pricing models (Jia et al., 2022a;b) study queueing systems with reusable resources, where price changes induce transient effects before the system reaches a new steady state. Furthermore, related models include tallying and weighted-tallying bandits (Malik et al., 2022; 2023),delayed-impact bandits (Tang et al., 2021), priming and wear-in/wear-out effects (Agrawal & Tulabandula, 2020), rebounding or satiation dynamics (Leqi et al., 2021), last-switch-dependent rewards (Laforgue et al., 2022; Foussoul et al., 2023), and bandits with deterministically evolving preferences (Khosravi et al., 2023). These works typically impose specific forms on how past actions affect rewards, using arm-wise history statistics or application-specific dynamics, such as recency-weighted play counts, congestion effects, or queueing transients. In contrast, we consider a general matrix-valued memory map $A(a_{t-M}, \ldots, a_{t-1})$ that accommodates order-sensitive and cross-action interactions without prescribing a particular memory mechanism, with regret bounds independent of the number of arms.

Generalized linear bandits. Generalized linear bandits model expected rewards through a known link function applied to a linear predictor. Early work developed optimistic algorithms based on maximum likelihood estimation (MLE) (Filippi et al., 2010; Li et al., 2017). For logistic ban-

dits, Faury et al. (2020) removed the worst case curvature dependence from the leading $\sqrt{T}$ term, while Abeille et al. (2021) refined the role of curvature through instance-dependent upper and lower bounds. Lee et al. (2024) developed likelihood-ratio confidence sequences for generalized linear models and applied them to bandits. Liu et al. (2024) studied self-concordance in natural exponential families and derived regret guarantees for generalized linear bandits with subexponential reward tails. Computational efficiency has also received considerable attention (Jun et al., 2017; Faury et al., 2022); in particular, Zhang et al. (2026) used online mirror descent in place of MLE to obtain efficient one-pass updates. Closely related to our update mechanism, Sawarni et al. (2024) studied generalized linear bandits with limited adaptivity, including rarely-switching algorithms. These works address nonlinear reward learning, but their regret guarantees do not account for the effects of current actions on future memory states. In contrast, the GLBM setting allows rewards to depend on the preceding M actions and compares the learner with the optimal complete action sequence. Our extension combines likelihood-based confidence sets with Bellman planning and rare switching. Here, rare switching not only reduces replanning frequency but also controls the memory-induced error at episode boundaries, allowing the finite memory structure to separate this cost from statistical estimation error.

B Detailed Algorithmic Steps in GLBM

In this appendix, we provide the omitted details of Algorithm 2 in GLBM. We begin with the action selection with frozen parameters in the algorithm.

Action selection with frozen parameters. Like Algorithm 1, RSM-GLMUCB freezes its parameter estimate at the beginning of each episode. For generalized rewards, however, we replace ridge regression with maximum likelihood estimation. Let $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_2 \le S\}$. At the beginning of episode k, we compute the maximum likelihood estimate $\widehat{\theta}_k$, construct a nested confidence set $\mathcal{C}_k$ of possible values of $\theta^\star$, and select the point $\overline{\theta}_k$ in $\mathcal{C}_k$ closest to $\widehat{\theta}_k$:

$$\begin{aligned}
&\widehat{\theta}_k \in \arg\min_{\theta\in\Theta} \mathcal{L}_{\tau_k-1}(\theta),\\
&\mathcal{C}_k = \mathcal{C}_{k-1} \cap \left\{\theta \in \Theta : \mathcal{L}_{\tau_k-1}(\theta) - \mathcal{L}_{\tau_k-1}(\widehat{\theta}_k) \le v c_T(\delta)\right\},\\
&\overline{\theta}_k = \arg\min_{\theta\in\mathcal{C}_k} \|\theta - \widehat{\theta}_k\|_2^2,
\end{aligned} \tag{9}$$

where $\mathcal{L}_t(\theta)$ measures how well θ explains the rewards observed in the first t rounds. It is obtained by multiplying the negative log-likelihood by v and omitting terms independent of θ, so minimizing it yields the maximum likelihood estimate. The confidence threshold $c_T(\delta)$ determines the allowed increase $vc_T(\delta)$ above this minimum when constructing the confidence set:

$$\mathcal{L}_t(\theta) = \sum_{j=1}^{t} \left[b(\phi(s_j,a_j)^\top\theta) - y_j\phi(s_j,a_j)^\top\theta\right], \tag{10}$$

$$c_T(\delta) = \log\left(\frac{2}{\delta}\right) + d + d\log\left(1 + \frac{4RST}{vd}\left[U_\mu RS + \frac{\sqrt{vU_\mu}}{\delta}\right]\right). \tag{11}$$

If the intersection in Equation (9) is empty, the algorithm sets $\mathcal{C}_k = \mathcal{C}_{k-1}$, so the confidence sets remain nested and nonempty across episodes. After constructing $\mathcal{C}_k$ and $\overline{\theta}_k$, at each round t of episode k the algorithm selects an action that maximizes the optimistic action value, as in Algorithm 1:

$$a_t \in \arg\max_{a\in\mathcal{A}} \{u_k(s_t,a) + \gamma J_k(f(s_t,a))\}, \tag{12}$$

where $u_k(s,a)$ is the optimistic reward estimate and $J_k(s)$ is the corresponding optimal discounted value, defined as

$$u_k(s,a) = \max_{\theta\in\mathcal{C}_k} \mu\left(\phi(s,a)^\top\theta\right), \quad J_k(s) = \sup_{\overline{a}_0,\overline{a}_1,\ldots} \sum_{j=0}^{\infty} \gamma^j u_k(\overline{s}_j,\overline{a}_j). \tag{13}$$

Here, $\overline{s}_0 = s$ and $\overline{s}_{j+1} = f(\overline{s}_j, \overline{a}_j)$. As in the linear model, this planning step accounts for both the immediate optimistic reward and the effect of current actions on future memory states. The algorithm plays a_t and observes the reward y_t. As discussed in Section 3, J_k can be approximated using discounted Howard policy iteration for small-scale problems or numerical planning methods for larger problems. In our semantic cache experiment (Section 6 and Appendix C), we further exploit its disjoint-bundle structure, combining a finite-vertex confidence approximation with sorting-based cache optimization and cyclic refreshes to implement the optimistic planning step more efficiently (Appendix C.5).

Determinant-triggered policy updates. Unlike Algorithm 1, RSM-GLMUCB also uses the local slope of the mean reward function when updating its information matrices. For the selected feature $\phi(s_t, a_t)$, it computes

$$w_t = \mu'(\phi(s_t, a_t)^\top \overline{\theta}_k)\mathbf{1}\left\{\rho\omega_k(\phi(s_t, a_t)) \le 1\right\}, \tag{14}$$

where $\omega_k(\phi(s,a)) = \max_{\theta\in\mathcal{C}_k} \phi(s,a)^\top\theta - \min_{\theta\in\mathcal{C}_k} \phi(s,a)^\top\theta$ denotes the prediction width for the feature $\phi(s,a)$. Under the property that $|\mu''(z)| \le \rho\mu'(z)$, the width condition ensures that the reward slopes associated with parameters in $\mathcal{C}_k$ differ by at most a constant factor. This allows the estimated slope to be used as a weight.

Finally, the algorithm updates V_t and H_t by adding $\phi(s_t, a_t)\phi(s_t, a_t)^\top$ and $w_t\phi(s_t, a_t)\phi(s_t, a_t)^\top$, respectively. The unweighted matrix V_t records all selected features, whereas H_t incorporates local slope information only when the prediction width is sufficiently small. In particular, when $w_t = 0$, H_t remains unchanged, whereas V_t continues to accumulate information and may still trigger a policy update. While Algorithm 1 monitors only V_t, RSM-GLMUCB starts a new episode when either $\det(V_t) \ge 2\det(\overline{V}_k)$ or $\det(H_t) \ge 2\det(\overline{H}_k)$. At the start of a new episode, the algorithm updates the confidence set, estimator, and policy, and resets both reference matrices.

C Experimental Details

This appendix provides additional details of our experiments. We first describe the planning algorithm used to approximate the value function $J_k(s)$ under small-scale settings.

C.1 Discounted Howard Policy Iteration

At the start of episode k, we use discounted Howard policy iteration to approximate $J_k(s)$ in Equation (6). The subroutine takes the frozen optimistic reward function $u = u_k$, discount factor $\gamma \in (0,1)$, and iteration budget $I_{\max}$. Since the memory transition f is known and deterministic, planning requires no additional interaction with the environment. The initial policy is myopic: it selects an action that maximizes the immediate optimistic reward,

$$\pi_0(s) \in \arg\max_{a\in\mathcal{A}} u(s,a). \tag{15}$$

Algorithm 3 Discounted Howard Policy Iteration at Episode k

1: **Input:** Reward function u, discount factor $\gamma \in (0,1)$, and iteration budget $I_{\max}$.
2: **Initialize:** Set π_0 according to Equation (15).
3: **for** $i = 0, \dots, I_{\max} - 1$ **do**
4: Update π_{i+1} according to Equation (16) with $\widetilde{Q}_i(s,a)$ and $\widetilde{J}_i(s)$ defined as in Equation (17).
5: **end for**
6: **return** $\widetilde{J}_{I_{\max}}$ defined as in Equation (17).

At iteration i, the algorithm updates the policy as

$$\pi_{i+1}(s) \in \arg\max_{a\in\mathcal{A}} \widetilde{Q}_i(s,a), \tag{16}$$

where $\widetilde{Q}_i(s,a)$ and $\widetilde{J}_i(s)$ denote the action value and the discounted value of policy π_i, respectively, defined as

$$\widetilde{Q}_i(s,a) = u(s,a) + \gamma\widetilde{J}_i(f(s,a)), \quad \widetilde{J}_i(s) = u(s,\pi_i(s)) + \gamma\widetilde{J}_i(f(s,\pi_i(s))). \tag{17}$$

Since $\gamma \in (0,1)$, $\widetilde{J}_i$ is uniquely determined by this Bellman equation. Unlike the myopic initialization, this update accounts for both the immediate optimistic reward and the value of the next memory state. The algorithm performs $I_{\max}$ policy improvements. It returns $\widetilde{J}_{I_{\max}}$ as an approximation to J_k, which is then used in the action-selection rules.

Theorem 5. *Under the conditions of Theorem 1, consider RSM-LinUCB with J_k in Equation (4) replaced at each episode by $\widetilde{J}_{I_{\max}}$ returned by Algorithm 3. Let π denote the resulting policy. Then,*

$$\begin{aligned}\mathrm{Reg}_T(\pi) \le &\left(4d\log\left(1+\frac{TR^2}{\lambda d}\right)+6\right)RSM + 4RSd\log\left(1+\frac{TR^2}{\lambda d}\right)\\ &+4\sqrt{Td\log\left(1+\frac{TR^2}{\lambda d}\right)}\left[\sigma\sqrt{d\log\left(1+\frac{TR^2}{\lambda d}\right)+2\log\left(\frac{1}{\delta}\right)}+\sqrt{\lambda}S\right]\\ &+2RST^2\left(1-\frac{1}{T}\right)^{I_{\max}+1},\end{aligned}$$

holds with probability at least $1-\delta$. Furthermore, if $I_{\max} \ge \lceil 2T\log T\rceil$, the additional term is at most $2RS$, and hence

$$\mathrm{Reg}_T(\pi) = \widetilde{O}\left(dRS(M+1)+\sigma d\sqrt{T}\right).$$

C.2 Synthetic Experimental Setup

Data processing and parameter selections. For the instance construction, we use the coordinate action set $\mathcal{A} = \{e_1,\dots,e_{10}\}$ with a fixed ordering. For a history $s = (a_{t-M},\dots,a_{t-1})$, the memory map is

$$A(s) = \begin{cases} Re_ie_i^\top, & \text{if } a_{t-M} = \cdots = a_{t-1} = e_i,\\ 0_{d\times d}, & \text{otherwise.}\end{cases}$$

Thus, after switching to an action, the learner must repeat it for M rounds before receiving a nonzero mean reward. The parameter is $\theta^\star = Su$, where u is a fixed nonnegative unit vector.

In the implementation, RSM-LinUCB uses regularization $\lambda = 0.625$ and confidence parameter $\delta = 0.05$, together with the discount factor specified in the main text. Optimistic rewards are clipped to $[-RS, RS]$. For the block baselines, the two OFUL variants use their respective episode-length rules; when the nominal informative length of OFUL (original) is nonpositive, we set it to 1.

Comparator. We use ten balanced initial histories constructed from cyclic shifts of a repeated, fixed random permutation of the actions. Each history contains all actions equally often, and each action appears last in one of the ten histories. Results are averaged over these histories within each noise seed. All methods share the same initial histories and paired noise sequences, and the parameter rules are unchanged across the four sweeps. The comparator $\mathrm{OPT}_T(\theta^\star)$ is thus the exact finite-horizon optimum from the same initial history. Let ℓ_i denote the length of the trailing run of e_i in this history, with $\ell_i = 0$ if the history does not end in e_i. Then

$$\mathrm{OPT}_T(\theta^\star) = RS\max_{i\in[d]} u_i(T-M+\ell_i)_+,$$

where $(x)_+ = \max\{x,0\}$. Each term is the cumulative mean reward obtained by repeating e_i throughout the horizon, including the initial zero-reward rounds. These constant action sequences suffice: active rewards are constant and nonnegative, and each switch requires M zero-reward rounds, so delaying a switch or switching repeatedly cannot improve upon either staying with the initial trailing action or switching immediately to another action.

Moreover, the same structure also allows us to compute J_k exactly by comparing the discounted returns of repeatedly selecting each action under u_k. We only need to track the last action and how many consecutive times it has been selected, capped at M, giving $O(dM)$ effective states on which the value function and its greedy policy are computed exactly.

C.3 Additional Synthetic Experiments

To complement the synthetic experiments in the main text, we consider an additive memory matrix of the form studied by Clerici et al. (2023); Ann et al. (2026). In this setting, the preceding actions contribute jointly to the memory matrix, and the learner selects actions from a continuous action set.

Data processing and parameter selections. We use $d = 10$, $\mathcal{A} = \{a \in \mathbb{R}^d : \|a\|_2 \le 1\}$, and independent Gaussian noise with standard deviation $\sigma = 0.25$. For a history $s_t = (a_{t-M}, \dots, a_{t-1})$, the memory map is

$$A(s_t) = \frac{R}{M+1}\left(I_d + \sum_{j=1}^{M} a_{t-j} a_{t-j}^\top\right).$$

The normalization ensures $\|A(s_t)\|_{\mathrm{op}} \le R$, allowing us to vary the memory length and operator-norm bound separately. We set $\theta^\star = Su$, using the same fixed nonnegative unit direction u as in the synthetic experiments in the main text. All methods start from zero initial memory and share paired noise sequences. Unless varied, $T = 10^6$, $M = 100$, and $R = S = 1$. We record a continuous trajectory up to $T = 10^6$ and separately vary $M \in \{100, 200, \dots, 1000\}$ and $R, S \in \{10, 20, \dots, 100\}$, one parameter at a time. RSM-LinUCB uses $\lambda = 0.625$, $\delta = 0.05$, and discount factor $\gamma = 0.999$, with optimistic rewards clipped to $[-RS, RS]$. We compare against the same four baselines as in the synthetic experiments in the main text. Results average 24 seeds; shading denotes one sample standard deviation.

Comparator. We measure cumulative pseudo-regret against the exact finite-horizon optimum from the same zero initial memory. At round t, the mean reward is at most $RS(1 + \min\{t-1, M\})/(M+1)$, and the constant action sequence $a_t = u$ attains this bound at every round. Thus,

$$\mathrm{OPT}_T(\theta^\star) = \frac{RS}{M+1}\sum_{t=1}^{T}\big(1 + \min\{t-1, M\}\big),$$

which equals $RS(T - M/2)$ for $T \ge M$, including the initial memory build-up. For the implementation of RSM-LinUCB, we follow the gradient-based optimization approach of Clerici et al. (2023); Ann et al. (2026) and approximate optimistic planning by projected gradient ascent over periodic action sequences of length M. We evaluate each sequence's discounted return from the current memory, including its infinite periodic tail, and use at most 60 iterations per initialization, with four initializations and a warm start from the previous sequence. The selected sequence is repeated until the next determinant trigger.

Experimental Results. As shown in Figure 2, RSM-LinUCB achieves the lowest mean regret at every tested value of M, R, and S. At $T = 10^6$, its mean regret is approximately 52.7% lower than that of the best baseline. These results complement the main experiments by demonstrating the advantage of our memory-aware planning approach under additive memory dynamics and a continuous action set.

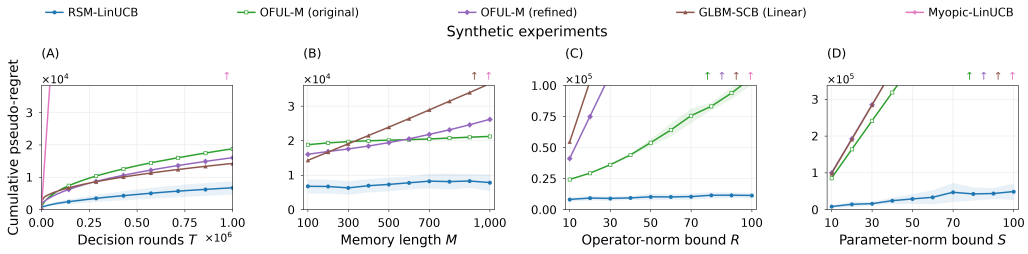

Figure 2: Cumulative pseudo-regret for additional synthetic experiments with additive memory. Shading denotes one sample standard deviation. Some baseline curves exceed the displayed range.

C.4 KV CACHE EXPERIMENTAL SETUP

Data processing and parameter selections. We partition the Mooncake conversation trace (Qin et al., 2025) into construction, calibration, and evaluation sets using a fixed, content-independent hash of its 10-second timestamp bins. Entire bins are assigned in proportions $40\%/20\%/40\%$, yielding 4,978, 2,230, and 4,823 requests, respectively. We allocate a larger construction share than in a preliminary $20\%/20\%/60\%$ split to identify more reusable prefixes and obtain a richer action catalogue. This is quite useful when studying longer memory horizons: a small catalogue can become largely covered by the retained actions, leaving little scope for comparing different cache management policies. The split determines the available data, whereas the tested memory lengths are chosen separately from the retention timescale. Using only construction requests, we retain prefixes observed at least twice and project each request onto its deepest retained prefix. Distinct projected endpoints define candidate refresh actions. Shared blocks count only once toward capacity and reward. We then contract nonbranching chains whose blocks always have identical availability, preserving the permitted endpoints and branch points. The resulting catalogue contains $K = 855$ actions and $d = 864$ informative segment coordinates. The catalogue and demand representation remain fixed across all tested values of M; blocks outside the retained catalogue contribute no reusable-token reward.

Rewards are divided by a fixed token scale $Z = 131{,}072$ and averaged over 36 independently sampled requests per round. For the reported experiments, we use the conservative request-level range $[0, 1]$, giving the sub-Gaussian noise parameter $\sigma = 1/(2\sqrt{36}) = 1/12$. The parameter norm bound is set to $S = 1.25\|\widehat{\theta}_{\mathrm{cal}}\|_2 \approx 0.1452$, where $\widehat{\theta}_{\mathrm{cal}}$ is the calibration mean expressed in the scaled feature coordinates. Our method uses $\lambda = d\sigma^2/S^2 \approx 284.72$ and confidence level 10^{-5}. The theoretical guarantee requires $\|\theta^\star\|_2 \le S$; we verify this condition for the fixed empirical evaluation distribution without exposing its demand weights to the learner. The baselines (OFUL-M, GLBM-SCB) retain their respective block-length rules, confidence interval, and estimation updates. Nonpositive prescribed informative lengths L are replaced by one. GLBM-SCB uses its ordered online mirror-descent updates, with the implemented initialization $\lambda_{\mathrm{GLBM}} = L/d$. For scalability, the baseline methods use the same directional tree-planning machinery described below. Myopic LinUCB maximizes the confidence score for current state without planning future cache effects.

Simulation protocol. Each round represents a 10-second cache control epoch. At the start of each round, the controller refreshes one catalogue prefix, making its blocks available for current and the following M rounds. Refreshing an already resident block renews its retention. The $M = 180$ setting therefore represents approximately 30 minutes of retention, motivated by Mooncake Store's documented soft-pin TTL (Mooncake Team, 2026). To keep cache management lightweight, reduce communication overhead, and limit exposure of individual access patterns, we model a management controller separate from the request-serving engine, motivated by aggregate cache monitoring and privacy concerns in KV-cache sharing (LMCache; Wu et al., 2025). The controller makes refresh decisions at the epoch level, while the engine handles per-request matching to KV blocks and exports only epoch-level reuse statistics rather than prompt contents, request prefix identifiers, or tensor payloads. This interface is motivated by the aggregate prefix-cache query and hit counters available in serving systems such as vLLM (vLLM Contributors). The controller knows the fixed action catalogue and its own refresh history, which suffice to compute the cache feature; aggregate rewards then allow it to estimate unknown demand without identifying each arriving request's KV blocks. This separation avoids maintaining request-level access logs at the controller while retaining the feedback needed for learning.

Let $p_i \in \{0, 1\}^d$ indicate the segments included in refresh action i, and let $g(s_t) \in \{0, 1\}^d$ indicate those still resident from the preceding M rounds. After action i_t, the available segments are indicated by $w_t = g(s_t) + (1 - g(s_t)) \odot p_{i_t}$: previously resident segments remain available, and the current action adds the missing segments. We use the scaled cache feature $x_t = w_t/\sqrt{d}$. Both w_t and x_t are known to the controller because they depend only on its refresh history and the fixed action catalogue. For each request Q, let $z_j(Q)$ be the number of matching prefix tokens in retained segment j. The number of reusable tokens is then $w_t^\top z(Q)$. Each round draws 36 requests and

returns their normalized average reuse,

$$Y_t = (36Z)^{-1} \sum_{j=1}^{36} w_t^\top z(Q_{t,j}),$$

where Z is the fixed normalization constant. Defining $\theta^\star = (\sqrt{d}/Z)\mathbb{E}_{\text{eval}}[z(Q)]$ gives $Y_t = x_t^\top \theta^\star + \varepsilon_t$, where ε_t has zero mean conditional on the past and the chosen action. The controller observes only Y_t; the individual demand vectors are used by the simulator to generate this feedback.

To express this model in the paper's LBM notation, we add one coordinate. Let $h = \max_i \|p_i\|_2^2$ and define

$$a_i = \frac{1}{\sqrt{1+h}} \begin{pmatrix} 1 \\ p_i \end{pmatrix}, \quad A(s) = \sqrt{\frac{1+h}{d}} \begin{pmatrix} 0 & g(s)^\top \\ 0 & \operatorname{diag}(1-g(s)) \end{pmatrix}.$$

These definitions ensure $\|a_i\|_2 \le 1$ and $A(s_t)^\top a_{i_t} = (0, x_t)$. With augmented parameter $\bar{\theta}^\star = (0, \theta^\star)$, the conditional mean reward is exactly $\langle a_{i_t}, A(s_t)\bar{\theta}^\star \rangle$. The added feature coordinate is always zero and is omitted from regression, leaving d informative coordinates.

Comparator. The comparator $\mathrm{OPT}_T(\theta^\star)$ is the maximum cumulative mean reward over all T-round refresh sequences in the fixed catalogue, starting from the same empty cache as the learner. We compute it offline using mean token demands under the empirical evaluation distribution.

Let $q = M + 1$, and let F_j denote the largest per-round mean reward obtainable from the union of at most j catalogue paths. These values are computed exactly by tree dynamic programming using the mean demand weights. To account for the initial cache build-up, we construct an ordered list by repeatedly adding the path with the largest additional demand coverage. For the reported catalogue, we verify against the exact dynamic program that the first j paths attain F_j for every $j \le q$, using integer token counts before normalization. Refreshing these paths in order and then cycling through them attains $F_{\min\{t,q\}}$ at every round. No refresh sequence can exceed this value at round t, since its cache contains the union of at most $\min\{t, q\}$ paths. Therefore, we set $\mathrm{OPT}_T(\theta^\star) = \sum_{t=1}^{T} F_{\min\{t,M+1\}}$ to denote the optimal cumulative reward over T rounds in the experiment. Given the mean demand weights, this offline computation takes $O(dq^3 + Kdq)$ time, including the tree-DP calls for all $j \le q$ and verification of the ordered paths.

Accelerated planning. In the KV cache experiment, we utilize the special prefix-cache structure to obtain a computationally efficient approximation to the optimistic planning step in RSM-LinUCB. Specifically, we approximate the ellipsoidal confidence bonus in u_k using a finite set of Gaussian directions to use tree dynamic programming for the linear objectives:

$$\widetilde{u}_k(x) = \mathrm{clip}_{[0,B]} \left(x^\top \overline{\theta}_k + \frac{\overline{\beta}_k}{c} \max_{1 \le j \le N} \left| g_{k,j}^\top L_k^\top x \right| \right), \tag{18}$$

where x is the informative cache feature, $L_k L_k^\top = \overline{V}_k^{-1}$, $g_{k,j} \sim \mathcal{N}(0, I_d)$ are independent directions drawn at each planning episode, and $B = RS$ is the known expected reward bound. We use $c = 1.5$ and choose $N = O(\log(T/\delta_{\text{plan}}))$ to control the additional failure probability.

The resulting score is the clipped maximum of $2N$ linear objectives, each assigning an additive weight to every tree segment. For each objective, a tree-knapsack dynamic program finds the best union of at most $q = M + 1$ prefix paths exactly, counting shared segments only once. We then approximate the planning policy by selecting the union with the highest score and cyclically refreshing its constituent paths until the next determinant trigger. This implementation leaves the estimator and update schedule unchanged, and the resident cache equals the selected union after at most M rounds. Each tree optimization takes $O(dq^2)$ time. Including matrix factorization and direction transformations, each replanning call takes $O(d^3 + Nd^2 + Ndq^2)$ time and $O(d^2 + Ndq)$ planning memory in the batched implementation.

As in polyhedral confidence set methods (Dani et al., 2008) and randomized linear bandits (Abeille & Lazaric, 2017), our approximation introduces only an additional $O(\sqrt{d})$ factor in the estimation error (the second term in Theorem 1), up to logarithmic terms. The ridge estimator and its confidence guarantee remain unchanged, and the additional cost only comes from the confidence bound used in

the optimistic reward estimate $\widetilde{u}_k$ in Equation (18). Let $x^\star$ denote an optimal steady cache feature. Each Gaussian direction has a constant probability of producing a confidence bound at least as large as the original one at $x^\star$. Taking the maximum over the prescribed number of independent directions therefore ensures $\widetilde{u}_k(x^\star) \geq (x^\star)^\top \theta^\star$ across episodes with high probability. Furthermore, Gaussian concentration also controls the enlargement of the confidence bound uniformly over cache features, including the feature x_k selected by maximizing $\widetilde{u}_k$. Together with optimism at $x^\star$, this gives

$$(x^\star - x_k)^\top \theta^\star \leq \widetilde{u}_k(x_k) - x_k^\top \theta^\star \leq \widetilde{O}(\sqrt{d})\, \overline{\beta}_k \|x_k\|_{\overline{V}_k^{-1}}.$$

Thus, the reward gap is still controlled by the original confidence width, enlarged by only a $\sqrt{d}$ factor. Summing these widths introduces only an additional $\widetilde{O}(\sqrt{d})$ factor in the estimation error. The memory transients under cyclic refreshes remain bounded by BM per episode, so the additive memory dependence is preserved.

C.5 Semantic Cache Experimental Setup

Data processing and parameter selections. We use the released GTE embeddings and GPT-4o-mini responses from vCache/SemBenchmarkLmArena. Exact-query deduplication leaves 56,438 queries, which we randomly divide into disjoint 20%/20%/60% construction, calibration, and evaluation splits. After excluding construction entries larger than 64 KiB, we partition the remaining 11,276 entries into $K = 64$ disjoint cache bundles by recursively splitting along the embedding coordinate with the largest variance at its median, giving $d = 65$ features. Each query is assigned to its nearest construction entry using cosine similarity, with routing fixed independently of which bundles are resident. Reuse is counted as successful only if the similarity is at least 0.9 and the two queries share the benchmark's semantic-equivalence label. These labels are available only to the evaluator and do not certify factual correctness. Bundle sizes include the query and response text and float32 embeddings. For the 64 MiB payload budget C, we require $(M+1)\max_i \mathrm{size}(i) \leq C$ for the memory length M, ensuring feasibility for every refresh history. The budget excludes search index and runtime overheads. For GLBM-SCB, we use blocks of length $M + L$, where $L = \max\{1, \lfloor\sqrt{MT}\rfloor - M\}$ as used in their paper (Ann et al., 2026).

Cache dynamics and generalized rewards. An action refreshes one bundle, which remains available in the current round and the following M rounds. Read hits do not extend retention, and the simulator performs no additional admissions or evictions. Let $g(s) \in \{0,1\}^K$ indicate the bundles refreshed in the preceding M rounds. For $1 \leq M < K$, using the standard basis $e_0, \ldots, e_K$ of $\mathbb{R}^{K+1}$, we define action set $\mathcal{A} = \{a_1, \ldots, a_K\}$, with action $a_i = \sqrt{\frac{M}{M+1}} e_0 + \frac{1}{\sqrt{M+1}} e_i$, and memory matrix

$$A(s)^\top = \begin{pmatrix} 0 & 0_{1\times K} \\ \sqrt{\frac{M+1}{M}}\, g(s) & \sqrt{M+1}\,\mathrm{diag}(\mathbf{1} - g(s)) \end{pmatrix}.$$

Thus, every action has unit norm, $\|A(s)\|_{\mathrm{op}} \leq R = \sqrt{M+1}$, and

$$\phi(s, a_i) = A(s)^\top a_i = \begin{pmatrix} 0 \\ g(s) + \mathrm{diag}(\mathbf{1} - g(s)) e_i^{(K)} \end{pmatrix},$$

where $e_i^{(K)}$ is the i-th standard basis vector of $\mathbb{R}^K$. The nonzero coordinates therefore record exactly which bundles are available after the refresh, giving $d = K + 1 = 65$.

The calibration split fixes the bundle arrival proportions ℓ_i, while the evaluation split determines the unknown conditional success probabilities p_i. Simulated queries first select a bundle according to ℓ_i and then sample a held-out query assigned to that bundle. Accordingly,

$$\theta^\star = (0, \ell_1 p_1, \ldots, \ell_K p_K)^\top,$$

and thus $S = \|(\ell_1, \ldots, \ell_K)\|_2 \approx 0.1256$. The learner observes only the resulting service time feedback. With unit rate Poisson arrivals and independently sampled queries, successful cache service has rate $z = \phi(s,a)^\top \theta^\star$, so interruptions have rate $1 - z$. Keeping the cache fixed during each service cycle gives $Y_t \sim \mathrm{Exp}(1-z)$, with $\mu(z) = 1/(1-z)$ and $v = 1$. All tested settings satisfy $RS < 1$, with M counting completed service cycles.

Comparator. The comparator $\mathrm{OPT}_T(\theta^\star)$ is the maximum cumulative mean reward over all T-round refresh sequences in the fixed catalogue, starting from the same empty cache as the learner and following the same retention rule. We compute it offline using the simulator's true bundle contributions $\ell_i p_i$, which are not revealed to the learner. Let $q = \min\{K, M+1\}$. An optimal sequence refreshes the q highest-contribution bundles in decreasing order and then cycles through them. At round t, any feasible sequence can keep at most $\min\{t, q\}$ bundles resident, and this sequence retains exactly the bundles with the largest contributions. Since μ is increasing, it therefore maximizes the mean reward at every round. Summing these rewards gives the exact finite-horizon optimum, including the initial cache build-up. Sorting the contributions and computing prefix sums takes $O(K \log K + q)$ time, without enumerating refresh sequences.

Accelerated planning. We exploit the disjoint-bundle structure to efficiently approximate the optimistic planning step in Algorithm 2. Specifically, we use a finite-vertex outer approximation for the confidence maximization in u_k and implement the resulting cache policy through cyclic refreshes. Following Dani et al. (2008), we approximate the optimistic reward $u_k(s,a)$ as

$$\widetilde{u}_k(s,a) = \mu\left(\mathrm{clip}_{[-RS,RS]}\left(\phi(s,a)^\top \overline{\theta}_k + \sqrt{d}\max_{0\le j\le K}\left|\phi(s,a)^\top F_k e_j\right|\right)\right),$$

where $F_k = \sqrt{8evc_T(\delta) + 4\lambda_2 S^2}\,\overline{H}_k^{-1/2}$ captures parameter uncertainty, $d = K+1$, and e_j denotes the j-th coordinate vector.

This approximation reduces optimistic cache optimization to $2d$ linear objectives. For each objective, the disjoint-bundle structure allows us to find its best q-bundle cache by selecting the $q = \min\{K, M+1\}$ largest bundle coordinates. Comparing these candidates solves the approximate optimistic optimization over q-bundle caches exactly. We then cyclically refresh the selected bundles until the next determinant trigger, establishing the selected cache after at most M initial rounds. Each replanning call takes $O(d^3)$ time for matrix factorization and $O(dK \log K)$ time for cache optimization, giving a total complexity of $O(d^3 + dK \log K)$.

As in the polyhedral confidence approximation of Dani et al. (2008) (Theorem 2), our implementation also introduces only an additional $O(\sqrt{d})$ factor in the estimation error (the second term in Theorem 3). The finite-vertex approximation contains the same $\overline{H}_k$-based confidence ellipsoid, so the resulting reward estimate remains optimistic, while its prediction confidence bound increases by at most $\sqrt{d}$ relative to that ellipsoid. Since the approximate cache optimization is solved exactly, the reward gap is controlled by this enlarged confidence bound without further error from numerical optimization. For the reciprocal link $\mu(z) = 1/(1-z)$, the relation

$$\mu(z') - \mu(z) \le \sqrt{U_\mu \mu'(z)}\,(z' - z),$$

for $-RS \le z \le z' \le RS$ allows the prediction error to be translated into a mean-reward difference using the local slope. The estimation error argument in Theorem 3 therefore applies with only the additional $O(\sqrt{d})$ factor. Importantly, curvature weighting still uses the original confidence-width bounds rather than the enlarged planning bound, so the curvature-induced error does not acquire this factor. The determinant triggers are unchanged, and the initial cache transitions still contribute at most $M\Delta_\mu$ per episode. Thus, the approximation enlarges only the estimation error term, while preserving the original orders of the memory-induced and curvature-induced errors.

D Detailed Proofs

Proof of Lemma 1. Fix any two memory states $s, s' \in \mathcal{S}$ and any action sequence $(a_0, a_1, \ldots)$. Let $(s_j)_{j\ge 0}$ and $(s'_j)_{j\ge 0}$ denote the resulting state trajectories, where $s_0 = s$, $s'_0 = s'$, and

$$s_{j+1} = f(s_j, a_j), \quad s'_{j+1} = f(s'_j, a_j).$$

Applying the same sequence of M actions causes the two memory states to coincide. Therefore, $s_j = s'_j$ for every $j \ge M$, and the rewards along the two trajectories may differ only during the first M rounds. Since every reward lies in $[-RS, RS]$, we obtain

$$\sum_{j=0}^{\infty}\gamma^j r_q(s_j,a_j) = \sum_{j=0}^{\infty}\gamma^j r_q(s'_j,a_j) + \sum_{j=0}^{M-1}\gamma^j\left[r_q(s_j,a_j) - r_q(s'_j,a_j)\right]$$

$$\leq J_q(s') + 2RS \sum_{j=0}^{M-1} \gamma^j.$$

Taking the supremum over all action sequences on the left-hand side gives

$$J_q(s) - J_q(s') \leq 2RS \sum_{j=0}^{M-1} \gamma^j \leq 2RSM.$$

Since s and s' are arbitrary, it follows that

$$\operatorname{span}(J_q) = \sup_{s\in\mathcal{S}} J_q(s) - \inf_{s\in\mathcal{S}} J_q(s) \leq 2RSM.$$

This proves the claimed bound. □

Proof of Lemma 2. Recall that

$$\beta_t = \sigma\sqrt{d\log\left(1+\frac{tR^2}{\lambda d}\right) + 2\log\left(\frac{1}{\delta}\right)} + \sqrt{\lambda}S, \quad \widehat{\theta}_t = V_t^{-1} b_t.$$

By the standard self-normalized concentration inequality, the event

$$\mathcal{E} \coloneqq \bigcap_{t=0}^{T} \left\{ \left\| \widehat{\theta}_t - \theta^\star \right\|_{V_t} \leq \beta_t \right\}$$

satisfies $\mathbb{P}(\mathcal{E}) \geq 1-\delta$. Throughout the proof, we work on the event $\mathcal{E}$. For every episode k and state–action pair (s,a), the Cauchy–Schwarz inequality gives

$$\left|\phi(s,a)^\top \left(\overline{\theta}_k - \theta^\star\right)\right| \leq \overline{\beta}_k \left\|\phi(s,a)\right\|_{\overline{V}_k^{-1}}.$$

Since $r(s,a) \in [-RS, RS]$, clipping the optimistic estimate to $[-RS, RS]$ preserves optimism. Therefore, $u_k(s,a) \geq r(s,a)$ holds for every k and (s,a). For any reward function $r_q : \mathcal{S}\times\mathcal{A} \to [-RS, RS]$, let J_q denote its optimal discounted value function and define

$$c_q \coloneqq \inf_{s\in\mathcal{S}} J_q(s).$$

By Lemma 1,

$$J_q(s) - c_q \in [0, 2RSM]$$

holds for every memory state $s \in \mathcal{S}$. After centering the Bellman optimality equation by c_q, we have

$$(1-\gamma)c_q + J_q(s) - c_q = \sup_{a\in\mathcal{A}} \left\{ r_q(s,a) + \gamma\big(J_q(f(s,a)) - c_q\big)\right\}.$$

Now take $r_q = r$. Consider any comparator action sequence $(a_1^\star, \ldots, a_T^\star)$ in Equation (2), and let $(s_t^\star)_{t=1}^{T+1}$ be its induced state trajectory. Evaluating the Bellman equation at $(s_t^\star, a_t^\star)$ gives

$$r(s_t^\star, a_t^\star) \leq\ (1-\gamma)c_{\theta^\star} + (J_{\theta^\star}(s_t^\star) - c_{\theta^\star}) - \gamma\left(J_{\theta^\star}(s_{t+1}^\star) - c_{\theta^\star}\right).$$

Summing over $t \in [T]$, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T} r(s_t^\star, a_t^\star) \\
\leq& T(1-\gamma)c_{\theta^\star} + (J_{\theta^\star}(s_1^\star) - c_{\theta^\star}) + (1-\gamma)\sum_{t=2}^{T}(J_{\theta^\star}(s_t^\star) - c_{\theta^\star}) - \gamma\left(J_{\theta^\star}(s_{T+1}^\star) - c_{\theta^\star}\right) \\
\leq& T(1-\gamma)c_{\theta^\star} + 2RSM\left[1 + (T-1)(1-\gamma)\right].
\end{aligned}$$

Since $\gamma = 1 - 1/T$, taking the supremum over comparator action sequences yields

$$\mathrm{OPT}_T(\theta^\star) \leq c_{\theta^\star} + 4RSM. \tag{19}$$

We now analyze the trajectory generated by RSM-LinUCB. Let

$$E_k \coloneqq \{\tau_k, \dots, \tau_{k+1} - 1\}$$

denote the set of rounds in episode k, where $\tau_{K+1} = T + 1$, and let

$$L_k \coloneqq |E_k| = \tau_{k+1} - \tau_k.$$

Define $c_k \coloneqq \inf_{s \in \mathcal{S}} J_k(s)$. Since $u_k(s, a) \in [-RS, RS]$, Lemma 1 also gives

$$J_k(s) - c_k \in [0, 2RSM]$$

for every $s \in \mathcal{S}$. By the action-selection rule in Algorithm 1 and the Bellman optimality equation, for every $t \in E_k$,

$$J_k(s_t^\pi) = u_k(s_t^\pi, a_t^\pi) + \gamma J_k(s_{t+1}^\pi).$$

Centering this equality by c_k gives

$$u_k(s_t^\pi, a_t^\pi) + \gamma \left(J_k(s_{t+1}^\pi) - c_k\right) = (1-\gamma)c_k + J_k(s_t^\pi) - c_k.$$

Summing over $t \in E_k$ yields

$$\begin{aligned}
&L_k(1-\gamma)c_k - \sum_{t \in E_k} u_k(s_t^\pi, a_t^\pi) \\
&= -\left(J_k(s_{\tau_k}^\pi) - c_k\right) - (1-\gamma) \sum_{t=\tau_k+1}^{\tau_{k+1}-1} (J_k(s_t^\pi) - c_k) + \gamma \left(J_k(s_{\tau_{k+1}}^\pi) - c_k\right) \\
&\le 2RSM.
\end{aligned}$$

On the event $\mathcal{E}$, optimism implies $J_k(s) \ge J_{\theta^\star}(s)$ for every $s \in \mathcal{S}$, and hence $c_k \ge c_{\theta^\star}$. Combining these observations, we obtain

$$\begin{aligned}
&L_k(1-\gamma)c_{\theta^\star} - \sum_{t \in E_k} r(s_t^\pi, a_t^\pi) \\
&\le L_k(1-\gamma)c_k - \sum_{t \in E_k} u_k(s_t^\pi, a_t^\pi) + \sum_{t \in E_k} \left[u_k(s_t^\pi, a_t^\pi) - r(s_t^\pi, a_t^\pi)\right] \\
&\le 2RSM + \sum_{t \in E_k} \left[u_k(s_t^\pi, a_t^\pi) - r(s_t^\pi, a_t^\pi)\right].
\end{aligned}$$

Finally, combining Equation (19) with the preceding inequality, summing over all episodes, and recalling that $\sum_{k=1}^K L_k = T$ and $1 - \gamma = 1/T$, we have

$$\begin{aligned}
\mathrm{Reg}_T(\pi) &= \mathrm{OPT}_T(\theta^\star) - \sum_{t=1}^T r(s_t^\pi, a_t^\pi) \\
&\le c_{\theta^\star} + 4RSM - \sum_{k=1}^K \frac{L_k}{T} c_{\theta^\star} + 2KRSM + \sum_{t=1}^T \left[u_{k(t)}(s_t^\pi, a_t^\pi) - r(s_t^\pi, a_t^\pi)\right] \\
&= 2(K+2)RSM + \sum_{t=1}^T \left[u_{k(t)}(s_t^\pi, a_t^\pi) - r(s_t^\pi, a_t^\pi)\right].
\end{aligned}$$

Since $\mathbb{P}(\mathcal{E}) \ge 1 - \delta$, this proves the bound. □

Proof of Theorem 1. Let $\mathcal{E}$ and β_t be as defined in the proof of Lemma 2, and recall that $\mathbb{P}(\mathcal{E}) \ge 1 - \delta$. Throughout the proof, we work on the same event $\mathcal{E}$ and write $B = RS$.

By Lemma 2, on $\mathcal{E}$,

$$\mathrm{Reg}_T(\pi) \le 2(K+2)BM + \sum_{t=1}^T \left[u_{k(t)}(s_t^\pi, a_t^\pi) - r(s_t^\pi, a_t^\pi)\right].$$

It therefore remains to bound the number of episodes and the cumulative reward-estimation error. The confidence bound used in the proof of Lemma 2, together with clipping to $[-B, B]$, gives

$$0 \le u_k(s,a) - r(s,a) \le 2\min\left\{B,\, \overline{\beta}_k \, \|\phi(s,a)\|_{\overline{V}_k^{-1}}\right\}$$

for every episode k and state–action pair (s,a). We first bound the total number of episodes. For every completed episode $k \in [K-1]$, the determinant-doubling condition implies

$$\det(\overline{V}_{k+1}) = \det(V_{\tau_{k+1}-1}) \ge 2\det(\overline{V}_k).$$

Since $\overline{V}_1 = V_0 = \lambda I_d$, it follows that

$$2^{K-1}\det(V_0) \le \det(\overline{V}_K) \le \det(V_T).$$

Therefore,

$$K \le 1 + \frac{1}{\log 2}\log\left(\frac{\det(V_T)}{\det(\lambda I_d)}\right).$$

Define

$$\Gamma_T \coloneqq \log\left(\frac{\det(V_T)}{\det(\lambda I_d)}\right) \le d\log\left(1 + \frac{TR^2}{\lambda d}\right) =: \overline{\Gamma}_T,$$

where the inequality follows from $\|\phi(s,a)\|_2 \le R$ and the arithmetic–geometric mean inequality applied to the eigenvalues of V_T. Thus,

$$K \le 1 + \frac{\overline{\Gamma}_T}{\log 2} \le 1 + 2\overline{\Gamma}_T. \tag{20}$$

We next compare the frozen information matrix with the current one. For any round t in episode k, the episode termination rule ensures that

$$\det(V_{t-1}) < 2\det(\overline{V}_k).$$

Since $V_{t-1} \succeq \overline{V}_k$, define

$$Q \coloneqq \overline{V}_k^{-1/2} V_{t-1} \overline{V}_k^{-1/2}.$$

Every eigenvalue of Q is at least one, while

$$\det(Q) = \frac{\det(V_{t-1})}{\det(\overline{V}_k)} < 2.$$

It follows that $Q \preceq 2I_d$, and therefore

$$V_{t-1} \preceq 2\overline{V}_k.$$

Equivalently, $\overline{V}_k^{-1} \preceq 2V_{t-1}^{-1}$, which implies

$$\|\phi(s_t^\pi, a_t^\pi)\|_{\overline{V}_k^{-1}} \le \sqrt{2}\, \|\phi(s_t^\pi, a_t^\pi)\|_{V_{t-1}^{-1}}. \tag{21}$$

Combining Lemma 2, the reward-estimation error bound, Equation (20) and (21), we obtain

$$\begin{aligned}
\mathrm{Reg}_T(\pi) &\le 2(K+2)BM + 2\sum_{t=1}^{T} \min\left\{B,\, \sqrt{2}\,\overline{\beta}_{k(t)}\, \|\phi(s_t^\pi, a_t^\pi)\|_{V_{t-1}^{-1}}\right\} \\
&\le \left(4d\log\left(1 + \frac{TR^2}{\lambda d}\right) + 6\right) BM + 2\sum_{t=1}^{T} \min\left\{B,\, \sqrt{2}\,\overline{\beta}_{k(t)}\, \|\phi(s_t^\pi, a_t^\pi)\|_{V_{t-1}^{-1}}\right\}.
\end{aligned} \tag{22}$$

It remains to control the cumulative confidence widths. By the elliptical potential lemma (Abbasi-Yadkori et al., 2011, Lemma 11),

$$\sum_{t=1}^{T} \min\left\{1,\, \|\phi(s_t^\pi, a_t^\pi)\|^2_{V_{t-1}^{-1}}\right\} \le 2d\log\left(1 + \frac{TR^2}{\lambda d}\right) = 2\overline{\Gamma}_T. \tag{23}$$

For notational convenience, let

$$q_t \coloneqq \|\phi(s_t^\pi, a_t^\pi)\|_{V_{t-1}^{-1}},$$

and define

$$\mathcal{I}_> \coloneqq \{t \in [T] : q_t > 1\}, \quad \mathcal{I}_\leq \coloneqq \{t \in [T] : q_t \leq 1\}.$$

By Equation (23), $|\mathcal{I}_>| \leq 2\overline{\Gamma}_T$. Moreover, since $\overline{\beta}_{k(t)} \leq \beta_T$, the Cauchy–Schwarz inequality gives

$$\begin{aligned}\sum_{t=1}^{T} \min\left\{B,\, \sqrt{2}\,\overline{\beta}_{k(t)} q_t\right\} &\leq B|\mathcal{I}_>| + \sqrt{2}\,\beta_T \sum_{t\in\mathcal{I}_\leq} q_t \\ &\leq 2B\overline{\Gamma}_T + \sqrt{2}\,\beta_T \sqrt{T \sum_{t\in\mathcal{I}_\leq} q_t^2} \\ &\leq 2B\overline{\Gamma}_T + 2\beta_T\sqrt{T\overline{\Gamma}_T}.\end{aligned}$$

Substituting this bound into Equation (22) yields

$$\begin{aligned}\mathrm{Reg}_T(\pi) \leq \left(4d\log\left(1+\frac{TR^2}{\lambda d}\right)+6\right)BM + 4Bd\log\left(1+\frac{TR^2}{\lambda d}\right) \\ + 4\sqrt{Td\log\left(1+\frac{TR^2}{\lambda d}\right)}\left[\sigma\sqrt{d\log\left(1+\frac{TR^2}{\lambda d}\right)+2\log\left(\frac{1}{\delta}\right)}+\sqrt{\lambda}S\right].\end{aligned}$$

Finally, recalling that $B = RS$, taking $\delta = T^{-2}$, and using $\sqrt{\lambda}S \leq \sigma\sqrt{d}$, we obtain

$$\mathrm{Reg}_T(\pi) = \widetilde{O}\left(dRS(M+1) + \sigma d\sqrt{T}\right).$$

Since all the bounds hold on the same event $\mathcal{E}$, the conclusion holds with probability at least $1-\delta$. This proves the claimed regret bound. □

Proof of Theorem 2. First, we consider the case when $M \geq 1$, and establish the memory-dependent lower bound. Let $e_1, \ldots, e_d$ denote the standard basis vectors in $\mathbb{R}^d$, and define the memory map

$$A(s) = \begin{cases} Re_i e_i^\top, & s = e_i^M \coloneqq (e_i, \ldots, e_i) \text{ for some } i \in [d], \\ 0, & \text{otherwise.}\end{cases}$$

Clearly, $\|A(s)\|_{\mathrm{op}} \leq R$ for every s. For each $i \in [d]$, let $\theta^{(i)} = Se_i$, and take the action set to be $\mathcal{A} = \mathbb{B}_2^d$. All environments, including the null environment, share the initial memory $s_1 = (0_d, \ldots, 0_d) \in \mathcal{A}^M$, which is also used by the comparator. For the first M rounds, the memory therefore contains at least one zero action, so $s_t \neq e_i^M$ for every $i \in [d]$ and all mean rewards are zero. The expected reward under parameter $\theta^{(i)}$ is then

$$r_{\theta^{(i)}}(s_t, a_t) = \begin{cases} RS(a_t)_i, & s_t = e_i^M, \\ 0, & \text{otherwise.}\end{cases}$$

For the environment $\theta^{(i)}$, an optimal comparator repeatedly selects e_i. Since the first M rounds yield zero reward, whereas every subsequent round yields reward RS, we have

$$\mathrm{OPT}_T\left(\theta^{(i)}\right) = RS(T-M).$$

Let $\mathrm{Reg}_T^{(i)}(\pi)$ denote the regret of policy π under parameter $\theta^{(i)}$, and let $\theta^{(0)} = 0$ denote the null environment. We couple all environments by using the same realization of the learner's internal randomness and the same noise sequence. Consider the trajectory generated by π in the null environment, and define

$$\tau_i = \inf\left\{t \in [T] : s_t = e_i^M\right\}.$$

If the set is empty, we set $\tau_i = \infty$. We further define the truncated hitting time

$$\overline{\tau}_i = \begin{cases} \tau_i, & \tau_i \leq T, \\ T+1, & \tau_i = \infty. \end{cases}$$

Before round $\overline{\tau}_i$, all conditional reward means are zero under both $\theta^{(0)}$ and $\theta^{(i)}$. Hence, under the above coupling, the two trajectories coincide up to the action selected at round $\overline{\tau}_i$. Since the expected reward in each subsequent round is at most RS, it follows pathwise that

$$\sum_{t=1}^{T} r_{\theta^{(i)}}(s_t^\pi, a_t^\pi) \le RS\,(T - \overline{\tau}_i + 1)\,.$$

Consequently,

$$\mathrm{Reg}_T^{(i)}(\pi) \ge RS\,(\overline{\tau}_i - (M+1))\,,$$

and therefore

$$\mathbb{E}\left[\mathrm{Reg}_T^{(i)}(\pi)\right] \ge RS\,\mathbb{E}\left[\overline{\tau}_i - (M+1)\right].$$

Now consider two distinct indices $i, j \in [d]$. If $\tau_i < \tau_j < \infty$, then the M actions immediately preceding τ_i are all equal to e_i, whereas those immediately preceding τ_j are all equal to e_j. Since $i \neq j$, these two blocks cannot overlap, and thus $\tau_j \ge \tau_i + M$. Let $\overline{\tau}_{(1)} \le \cdots \le \overline{\tau}_{(d)}$ denote the order statistics of $\overline{\tau}_1, \ldots, \overline{\tau}_d$. Since $\overline{\tau}_{(1)} \ge M+1$, the preceding separation property implies

$$\overline{\tau}_{(q)} - (M+1) \ge \min\{(q-1)M,\, T - M\}$$

for every $q \in [d]$. Therefore,

$$\begin{aligned}\frac{1}{d}\sum_{i=1}^{d}\mathbb{E}\left[\mathrm{Reg}_T^{(i)}(\pi)\right] &\ge \frac{RS}{d}\sum_{q=1}^{d}\min\{(q-1)M,\, T-M\} \\ &= \frac{RS}{d}\sum_{q=0}^{d-1}\min\{qM,\, T-M\}\,. \end{aligned} \tag{24}$$

Under the assumption $T \ge dM$, we have $qM \le T - M$ for every $q \in \{0, \ldots, d-1\}$. Hence,

$$\frac{1}{d}\sum_{i=1}^{d}\mathbb{E}\left[\mathrm{Reg}_T^{(i)}(\pi)\right] \ge \frac{(d-1)RSM}{2}.$$

It follows that

$$\sup_{\theta^\star, A}\mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \ge \frac{1}{d}\sum_{i=1}^{d}\mathbb{E}\left[\mathrm{Reg}_T^{(i)}(\pi)\right] \ge \frac{(d-1)RSM}{2}. \tag{25}$$

For $M = 0$, take $A = RI_d$. The problem then reduces to a standard linear bandit over $\mathbb{B}_2^d$, with parameter $R\theta^\star$ satisfying $\|R\theta^\star\|_2 \le RS$. Applying Proposition 2 of Zhang et al. (2025) with their scale parameter set to $RS/4$, we obtain

$$\sup_{\theta^\star, A}\mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \ge cdRS$$

for some universal constant $c > 0$, provided that $T \ge d$. Combining the two cases, we then obtain

$$\sup_{\theta^\star, A}\mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \ge \Omega\left(dRS(M+1)\right), \tag{26}$$

for every $M \ge 0$. We next establish the standard linear-bandit lower bound. Let the noise variables be independent Gaussian random variables,

$$\varepsilon_t \sim \mathcal{N}(0, \sigma^2),$$

and consider the memory-independent map $A(s) = RI_d$ for every $s \in \mathcal{A}^M$. This reduces the problem to a stochastic linear bandit over the unit Euclidean ball with effective parameter $R\theta^\star$. Applying Theorem 24.2 of Lattimore & Szepesvári (2020) after rescaling the observations by σ, there exists a parameter $\theta^\star$ satisfying $\|\theta^\star\|_2^2 = \frac{\sigma^2 d^2}{48TR^2}$ such that

$$\mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \ge \frac{\sigma d\sqrt{T}}{16\sqrt{3}}.$$

The parameter is admissible whenever $RS \geq \sigma d/\sqrt{48T}$. Therefore,

$$\sup_{\theta^\star, A} \mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \geq \frac{\sigma d\sqrt{T}}{16\sqrt{3}}. \tag{27}$$

Combining Equation (26), we conclude that, for every policy π,

$$\sup_{\theta^\star, A} \mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \geq \Omega\left(dRS(M+1) + \sigma d\sqrt{T}\right).$$

Taking the infimum over all policies proves Theorem 2. □

Proof of Theorem 3. We work on the event $\mathcal{E}$ from Lemma 3 which happens with probability at least $1-\delta$. According to Lemma 5, it remains to bound the reward estimation error.

Consider first a round t with $w_t > 0$, and let $k = k(t)$ denote its episode. The definition of w_t and Equation (37) in Appendix E imply that $\mu'(\phi_t^\top \theta) \leq \mathrm{e} w_t$ for every $\theta \in \mathcal{C}_k$. Since $\theta^\star$ and the maximizer defining $u_k(s_t, a_t)$ belong to the convex set $\mathcal{C}_k$, the mean value theorem and Lemma 4 give that

$$u_k(s_t, a_t) - r(s_t, a_t) \leq \mathrm{e}\, w_t \omega_k(\phi_t) \leq \sqrt{2}\,\mathrm{e}\,\beta_2 w_t \|\phi_t\|_{H_{t-1}^{-1}}.$$

Further denote that $\ell_t = w_t \|\phi_t\|^2_{H_{t-1}^{-1}}$. Applying the elliptical potential lemma (Lemma 11 in Abbasi-Yadkori et al. (2011)) to the vectors $\sqrt{w_t}\phi_t$ yields

$$\sum_{t=1}^{T} \min\{1, \ell_t\} \leq 2 \log\left(\frac{\det H_T}{\det H_0}\right) \leq 2\Gamma_2.$$

Therefore, by the Cauchy–Schwarz inequality and $\sum_{t=1}^T w_t \leq T U_\mu$,

$$\begin{aligned}\sum_{\substack{t: w_t > 0 \\ \ell_t \leq 1}} \left[u_{k(t)}(s_t, a_t) - r(s_t, a_t)\right] &\leq \sqrt{2}\,\mathrm{e}\,\beta_2 \sqrt{\left(\sum_{t=1}^{T} w_t\right)\left(\sum_{t: \ell_t \leq 1} \ell_t\right)} \\ &\leq 2\mathrm{e}\,\beta_2 \sqrt{U_\mu T \Gamma_2}.\end{aligned}$$

The determinant identity $\sum_{t=1}^T \log(1+\ell_t) = \log(\det H_T / \det H_0)$ also implies that at most $\Gamma_2 / \log 2$ rounds have $\ell_t > 1$. Bounding their estimation error by Δ_μ gives

$$\sum_{t: w_t > 0} \left[u_{k(t)}(s_t, a_t) - r(s_t, a_t)\right] \leq 2\mathrm{e}\,\beta_2 \sqrt{U_\mu T \Gamma_2} + \frac{\Delta_\mu \Gamma_2}{\log 2}. \tag{28}$$

We next consider rounds with $w_t = 0$. There are no such rounds if $\rho = 0$. Otherwise, Lemma 4 gives

$$1 < \rho \omega_{k(t)}(\phi_t) \leq \sqrt{2}\rho\beta_1 \|\phi_t\|_{V_{t-1}^{-1}} \text{ and } \|\phi_t\|^2_{V_{t-1}^{-1}} > \frac{1}{2\rho^2\beta_1^2}.$$

Using the determinant identity for V_t,

$$|\{t : w_t = 0\}| \leq \frac{\sum_{t=1}^{T} \log\left(1 + \|\phi_t\|^2_{V_{t-1}^{-1}}\right)}{\log\left(1 + 1/(2\rho^2\beta_1^2)\right)} \leq \frac{\Gamma_1}{\log\left(1 + 1/(2\rho^2\beta_1^2)\right)} \leq (1 + 2\rho^2\beta_1^2)\Gamma_1,$$

where the last inequality follows from $\log(1+x) \geq x/(1+x)$ for $x \geq 0$. Note that the final bound also holds when $\rho = 0$. Hence,

$$\sum_{t:\, w_t = 0} \left[u_{k(t)}(s_t, a_t) - r(s_t, a_t)\right] \leq \Delta_\mu (1 + 2\rho^2\beta_1^2)\Gamma_1. \tag{29}$$

Combining Lemma 5 with Equation (28) and (29), we obtain

$$\begin{aligned}
&\mathrm{Reg}_T(\text{RSM-GLMUCB}) \\
\leq &\left(3 + \frac{\Gamma_1 + \Gamma_2}{\log 2}\right) M\Delta_\mu + 2\mathrm{e}\,\beta_2\sqrt{U_\mu T\Gamma_2} + \frac{\Delta_\mu\Gamma_2}{\log 2} + \Delta_\mu(1 + 2\rho^2\beta_1^2)\Gamma_1 \qquad (30)\\
= &\widetilde{O}\left(d(M+1)\Delta_\mu + d\sqrt{vU_\mu T} + d^2\kappa v\rho^2\Delta_\mu\right), \qquad (31)
\end{aligned}$$

which completes the proof. □

Proof of Theorem 4. By Lemma 6, there exists an admissible action set and a common initial memory state for which every policy incurs expected regret at least $c_1\frac{d(M+1)\Delta_\mu}{1+\log(1+\rho RS)}$ for some universal constant $c_1 > 0$. On the other hand, Lemma 7 gives an admissible action set for which every policy incurs expected regret at least $c_2 d\sqrt{vU_\mu T}$, with a universal constant $c_2 > 0$.

For the fixed problem parameters, choose the construction with the larger of the two lower bounds, and we obtain that

$$\begin{aligned}
\inf_\pi \sup_{\theta^\star, A} \mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] &\geq \max\left\{c_1\frac{d(M+1)\Delta_\mu}{1+\log(1+\rho RS)},\ c_2 d\sqrt{vU_\mu T}\right\} \\
&\geq \frac{\min\{c_1, c_2\}}{2}\left(\frac{d(M+1)\Delta_\mu}{1+\log(1+\rho RS)} + d\sqrt{vU_\mu T}\right),
\end{aligned}$$

which proves the theorem. □

Proof of Theorem 5. We first control the error introduced by Algorithm 3. Fix an episode k and use $\widetilde{J}_i$ to denote the value function computed at iteration i within this episode. Recall that J_k is the optimal discounted value function under the frozen optimistic reward u_k. Since $\widetilde{J}_i$ is the value of policy π_i under the same reward function, it holds that $\widetilde{J}_i(s) \leq J_k(s)$ for every memory state $s \in \mathcal{S}$. By the contraction property of discounted Howard policy iteration (Lemma 2 in Scherrer (2013)),

$$\left\|J_k - \widetilde{J}_i\right\|_\infty \leq \gamma^i \left\|J_k - \widetilde{J}_0\right\|_\infty.$$

Since $u_k(s,a) \in [-RS, RS]$, both J_k and $\widetilde{J}_0$ take values in $[-RS/(1-\gamma), RS/(1-\gamma)]$. Therefore,

$$\left\|J_k - \widetilde{J}_i\right\|_\infty \leq \frac{2RS\gamma^i}{1-\gamma}. \qquad (32)$$

Within episode k, the implemented algorithm selects

$$a_t^\pi \in \arg\max_{a\in\mathcal{A}}\left\{u_k(s_t^\pi, a) + \gamma\widetilde{J}_{I_{\max}}(f(s_t^\pi, a))\right\}.$$

By the Bellman optimality equation for J_k and Equation (32),

$$\begin{aligned}
J_k(s_t^\pi) &= \max_{a\in\mathcal{A}}\left\{u_k(s_t^\pi, a) + \gamma J_k(f(s_t^\pi, a))\right\} \\
&\leq \max_{a\in\mathcal{A}}\left\{u_k(s_t^\pi, a) + \gamma\widetilde{J}_{I_{\max}}(f(s_t^\pi, a))\right\} + \frac{2RS\gamma^{I_{\max}+1}}{1-\gamma} \\
&= u_k(s_t^\pi, a_t^\pi) + \gamma\widetilde{J}_{I_{\max}}(s_{t+1}^\pi) + \frac{2RS\gamma^{I_{\max}+1}}{1-\gamma} \\
&\leq u_k(s_t^\pi, a_t^\pi) + \gamma J_k(s_{t+1}^\pi) + \frac{2RS\gamma^{I_{\max}+1}}{1-\gamma},
\end{aligned}$$

where the last inequality uses $\widetilde{J}_{I_{\max}} \leq J_k$. Consequently,

$$0 \leq J_k(s_t^\pi) - u_k(s_t^\pi, a_t^\pi) - \gamma J_k(s_{t+1}^\pi) \leq \frac{2RS\gamma^{I_{\max}+1}}{1-\gamma}. \qquad (33)$$

The first inequality follows from Bellman optimality. Thus, approximate planning introduces only the error on the right-hand side into the Bellman equality used in the proof of Lemma 2.

We now work on the confidence event $\mathcal{E}$ defined in the proof of Lemma 2. The same concentration argument applies to the trajectory of the implemented algorithm, so $\mathbb{P}(\mathcal{E}) \geq 1-\delta$. Let E_k, L_k, and $c_k = \inf_{s\in\mathcal{S}} J_k(s)$ be as defined in that proof. By Lemma 1,

$$J_k(s) - c_k \in [0, 2RSM].$$

holds for every $s \in \mathcal{S}$. Centering Equation (33) by c_k and summing over $t \in E_k$ yields

$$\begin{aligned}
&L_k(1-\gamma)c_k - \sum_{t\in E_k} u_k(s_t^\pi, a_t^\pi) \\
\leq &-\left(J_k(s_{\tau_k}^\pi) - c_k\right) - (1-\gamma)\sum_{t=\tau_k+1}^{\tau_{k+1}-1}(J_k(s_t^\pi) - c_k) + \gamma\left(J_k(s_{\tau_{k+1}}^\pi) - c_k\right) + \frac{2RSL_k\gamma^{I_{\max}+1}}{1-\gamma} \\
\leq &2RSM + \frac{2RSL_k\gamma^{I_{\max}+1}}{1-\gamma}.
\end{aligned}$$

On $\mathcal{E}$, optimism still gives $u_k(s,a) \geq r(s,a)$, and hence $J_k(s) \geq J_{\theta^\star}(s)$ and $c_k \geq c_{\theta^\star}$. Therefore,

$$L_k(1-\gamma)c_{\theta^\star} - \sum_{t\in E_k} r(s_t^\pi, a_t^\pi) \leq 2RSM + \frac{2RSL_k\gamma^{I_{\max}+1}}{1-\gamma} + \sum_{t\in E_k}\left[u_k(s_t^\pi, a_t^\pi) - r(s_t^\pi, a_t^\pi)\right].$$

Combining this inequality with Equation (19), summing over episodes, and recalling that $\sum_{k=1}^K L_k = T$ and $T(1-\gamma) = 1$, we obtain

$$\mathrm{Reg}_T(\pi) \leq 2(K+2)RSM + \sum_{t=1}^T\left[u_{k(t)}(s_t^\pi, a_t^\pi) - r(s_t^\pi, a_t^\pi)\right] + \frac{2RST\gamma^{I_{\max}+1}}{1-\gamma}. \tag{34}$$

The determinant and confidence width arguments in the proof of Theorem 1 remain unchanged. Using $\overline{\Gamma}_T$ and β_T defined there, Equation (34) gives

$$\mathrm{Reg}_T(\pi) \leq (4\overline{\Gamma}_T + 6)RSM + 4RS\overline{\Gamma}_T + 4\beta_T\sqrt{T\overline{\Gamma}_T} + \frac{2RST\gamma^{I_{\max}+1}}{1-\gamma}.$$

Substituting the definitions of $\overline{\Gamma}_T$ and β_T, and using $\gamma = 1 - 1/T$, proves the first bound.

Finally, if $I_{\max} \geq \lceil 2T\log T\rceil$, then

$$2RST^2\left(1-\frac{1}{T}\right)^{I_{\max}+1} \leq 2RST^2\exp\left(-\frac{I_{\max}+1}{T}\right) \leq 2RS.$$

This additional term is absorbed into $\widetilde{O}(dRS(M+1))$. The claimed regret order therefore follows from Theorem 1. All bounds hold on $\mathcal{E}$, completing the proof. □

E TECHNICAL LEMMAS

In this appendix, we introduce the technical lemmas and their proofs used in our paper.

Lemma 3 (Likelihood confidence for Algorithm 2)**.** *Under the generalized reward model, the event*

$$\mathcal{E} := \bigcap_{t=0}^T\left\{\mathcal{L}_t(\theta^\star) - \min_{\theta\in\Theta}\mathcal{L}_t(\theta) \leq vc_T(\delta)\right\}$$

satisfies $\mathbb{P}(\mathcal{E}) \geq 1-\delta$. On $\mathcal{E}$, the confidence sets in Equation (9) contain $\theta^\star$ at every episode, and $0 \leq u_k(s,a) - r(s,a) \leq \Delta_\mu$ for every k and $(s,a) \in \mathcal{S}\times\mathcal{A}$.

Proof. Firstly, we want to apply Theorem 3.1 in Lee et al. (2024) to the normalized negative log-likelihood $\mathcal{L}_t/v$.

To use this theorem, we first note that the unit-norm condition is satisfied by rescaling ϕ_t to ϕ_t/R and θ to $R\theta$. Hence, we only need to verify that $c_T(\delta)$ is a sufficient threshold. The conditional variance formula and the Cauchy–Schwarz inequality give $\mathbb{E}\left[\left|y_t - \mu(\phi_t^\top \theta^\star)\right|\right] \le \sqrt{vU_\mu}$. Hence, by Markov's inequality, with probability at least $1-\delta/2$, $\sum_{t=1}^{T}\left|y_t - \mu(\phi_t^\top\theta^\star)\right| \le \frac{2T\sqrt{vU_\mu}}{\delta}$ holds. On this event, for every $t \le T$,

$$\begin{aligned}\sup_{\theta\in\Theta}\left\|\nabla\left(\mathcal{L}_t/v\right)(\theta)\right\|_2 &\le \frac{R}{v}\left(\sum_{j=1}^{t}\left|y_j - \mu(\phi_j^\top\theta^\star)\right| + 2tU_\mu RS\right)\\ &\le \frac{2RT}{v}\left(U_\mu RS + \frac{\sqrt{vU_\mu}}{\delta}\right) =: G_T.\end{aligned}$$

Applying Theorem 3.1 in Lee et al. (2024) with confidence level $\delta/2$ and intersecting the two events yields, for all $t \le T$, with probability at least $1-\delta$,

$$\begin{aligned}\frac{\mathcal{L}_t(\theta^\star) - \min_{\theta\in\Theta}\mathcal{L}_t(\theta)}{v} &\le \log\left(\frac{2}{\delta}\right) + d\log\left(\max\left\{\mathrm{e}, \frac{2\mathrm{e}SG_T}{d}\right\}\right)\\ &\le \log\left(\frac{2}{\delta}\right) + d + d\log\left(1 + \frac{2SG_T}{d}\right) = c_T(\delta).\end{aligned}$$

This proves $\mathbb{P}(\mathcal{E}) \ge 1-\delta$. Since $\theta^\star \in \mathcal{C}_0$, the confidence set update then gives $\theta^\star \in \mathcal{C}_k$ by induction, and thus no intersection is empty. The remaining claim follows from the definition of u_k and the reward range $[\mu(-RS), \mu(RS)]$. □

Lemma 4 (Prediction widths)**.** *Define* $\beta_1^2 = 8\kappa v c_T(\delta) + 4\lambda_1 S^2$, *and* $\beta_2^2 = 8\mathrm{e}\, v c_T(\delta) + 4\lambda_2 S^2$. *Then on* $\mathcal{E}$*, for every round* t *in episode* k*, it holds that*

$$\omega_k(\phi_t) \le \sqrt{2}\beta_1\|\phi_t\|_{V_{t-1}^{-1}} \text{ and } \omega_k(\phi_t) \le \sqrt{2}\beta_2\|\phi_t\|_{H_{t-1}^{-1}}.$$

Proof. Fix an episode k and any $\theta, \theta' \in \mathcal{C}_k$. By Equation (9) and the optimality of $\widehat{\theta}_k$,

$$\frac{\mathcal{L}_{\tau_k-1}(\theta) + \mathcal{L}_{\tau_k-1}(\theta')}{2} - \mathcal{L}_{\tau_k-1}\left(\frac{\theta+\theta'}{2}\right) \le v c_T(\delta). \tag{35}$$

For every $\zeta \in \Theta$, we have

$$\nabla^2\mathcal{L}_{\tau_k-1}(\zeta) = \sum_{j<\tau_k}\mu'(\phi_j^\top\zeta)\phi_j\phi_j^\top \succeq \frac{1}{\kappa}\sum_{j<\tau_k}\phi_j\phi_j^\top.$$

The midpoint inequality for a convex function with this Hessian lower bound therefore gives

$$\frac{1}{8\kappa}\sum_{j<\tau_k}\left(\phi_j^\top(\theta-\theta')\right)^2 \le v c_T(\delta).$$

Since $\|\theta-\theta'\|_2 \le 2S$, it holds that

$$\|\theta-\theta'\|_{\overline{V}_k}^2 \le 8\kappa v c_T(\delta) + 4\lambda_1 S^2 = \beta_1^2. \tag{36}$$

For the weighted matrix, integrating $|(\log\mu')'| \le \rho$ gives

$$\exp(-\rho|z-z'|) \le \frac{\mu'(z)}{\mu'(z')} \le \exp(\rho|z-z'|), \tag{37}$$

for $z, z' \in [-RS, RS]$. The sets $\mathcal{C}_k$ are convex and nested. Thus, for every $j < \tau_k$, both $\zeta \in \mathcal{C}_k$ and $\overline{\theta}_{k(j)}$ belong to $\mathcal{C}_{k(j)}$. Whenever $w_j > 0$,

$$\rho\left|\phi_j^\top(\zeta - \overline{\theta}_{k(j)})\right| \le \rho\omega_{k(j)}(\phi_j) \le 1.$$

By Equation (37), $\mu'(\phi_j^\top \zeta) \geq \mathrm{e}^{-1} w_j$. The same inequality holds when $w_j = 0$. Therefore, along the segment joining θ and θ',

$$\nabla^2 \mathcal{L}_{\tau_k - 1}(\zeta) \succeq \mathrm{e}^{-1} \sum_{j < \tau_k} w_j \phi_j \phi_j^\top .$$

Applying the midpoint inequality to Equation (35) again leads to

$$\frac{1}{8\mathrm{e}} \sum_{j < \tau_k} w_j \left(\phi_j^\top (\theta - \theta')\right)^2 \leq v c_T(\delta),$$

and hence

$$\|\theta - \theta'\|_{\overline{H}_k}^2 \leq 8\mathrm{e}\, v c_T(\delta) + 4\lambda_2 S^2 = \beta_2^2. \tag{38}$$

This bound uses the fixed historical weights, rather than weights recomputed from the current estimate.

At any round t in episode k, both determinant ratios are less than two. By Lemma 12 in Abbasi-Yadkori et al. (2011), it holds that

$$\overline{V}_k^{-1} \preceq 2V_{t-1}^{-1}, \text{ and } \overline{H}_k^{-1} \preceq 2H_{t-1}^{-1}.$$

Combining these inequalities with Equation (36) and (38) and applying the Cauchy–Schwarz inequality proves the given width bounds. □

Lemma 5 (Episodes and regret)**.** *Let $\gamma = 1 - 1/T$. Further define that $\Gamma_1 = d \log\left(1 + \frac{TR^2}{\lambda_1 d}\right)$ and $\Gamma_2 = d \log\left(1 + \frac{TU_\mu R^2}{\lambda_2 d}\right)$. Then the number of episodes K satisfies $K \leq 1 + (\Gamma_1 + \Gamma_2)/\log 2$. Moreover, conditioned on $\mathcal{E}$, it holds that*

$$\mathrm{Reg}_T \leq (K+2) M \Delta_\mu + \sum_{t=1}^{T} \left[u_{k(t)}(s_t, a_t) - r(s_t, a_t)\right].$$

Proof. Since $\|\phi_t\|_2 \leq R$ and $0 \leq w_t \leq U_\mu$, the determinant–trace inequality gives

$$\log\left(\frac{\det V_T}{\det V_0}\right) \leq \Gamma_1 \text{ and } \log\left(\frac{\det H_T}{\det H_0}\right) \leq \Gamma_2.$$

Note that both matrices are nondecreasing in the positive semidefinite order. Each episode before the last ends with at least one determinant doubling, thus

$$(K-1)\log 2 \leq \log\left(\frac{\det V_T}{\det V_0}\right) + \log\left(\frac{\det H_T}{\det H_0}\right) \leq \Gamma_1 + \Gamma_2.$$

For the regret bound, we use the argument in the proof of Lemma 2. Once optimism holds, that argument uses only the reward range, the finite-memory transition and exact discounted planning within each episode. Furthermore, both r and u_k lie in $[\mu(-RS), \mu(RS)]$. Applying the positive affine transformation

$$h \to \frac{2RS}{\Delta_\mu}\left(h - \frac{\mu(RS) + \mu(-RS)}{2}\right)$$

to the rewards maps this interval to $[-RS, RS]$ and preserves the maximizing policies. Regret and reward-estimation errors are both multiplied by $2RS/\Delta_\mu$. Dividing the bound from the proof of Lemma 2 by this factor gives the decomposition in Lemma 5. □

Lemma 6. *For $d \geq 3$ and $T \geq (d-1)\max\{M, 1\}$, there exists an admissible action set such that, for every policy π,*

$$\sup_{\theta^\star} \mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \geq \Omega\left(\frac{d(M+1)\Delta_\mu}{1 + \log(1 + \rho RS)}\right).$$

Proof. Let $L = RS$ and $p = d - 1$. We construct p parameter instances that share the same action set and the same known memory matrix. In instance i, the learner receives a larger mean reward only after reaching a memory state associated with action i. Until that state is reached, the observations do not distinguish instance i from a common null environment. We first choose the two reward levels used in this construction and relate their difference to Δ_μ. Define

$$\Gamma_\mu = \sup_{0<h<2L} \left\{ \mu(-L+h) - \mu\left(-L + \frac{3h}{4}\right) \right\}.$$

We will show that Γ_μ is at least a logarithmic fraction of Δ_μ. Since $\mu'(z) > 0$ and $|\mu''(z)| \le \rho\mu'(z)$, for any $z, z' \in [-L, L]$,

$$|\log \mu'(z) - \log \mu'(z')| = \left| \int_{z'}^{z} \frac{\mu''(u)}{\mu'(u)}\, du \right| \le \rho |z - z'|.$$

Exponentiating both sides gives

$$e^{-\rho|z-z'|}\mu'(z') \le \mu'(z) \le e^{\rho|z-z'|}\mu'(z'). \tag{39}$$

Set $h_j = 2L(3/4)^j$ for $j \ge 0$, and let N be the smallest nonnegative integer such that $\rho h_N \le 1$ ($N = 0$ when $\rho = 0$). Since $h_{j+1} = 3h_j/4$, the intervals $[-L + 3h_j/4, -L + h_j]$, $j = 0, \dots, N$, are adjacent and together cover $[-L + 3h_N/4, L]$. Thus,

$$\Delta_\mu = \sum_{j=0}^{N} \left[\mu(-L+h_j) - \mu\left(-L + \frac{3h_j}{4}\right) \right] + \mu\left(-L + \frac{3h_N}{4}\right) - \mu(-L).$$

It remains to bound the change in μ over the uncovered interval $[-L, -L + 3h_N/4]$. For any $x \in [-L, -L + 3h_N/4]$ and $y \in [-L + 3h_N/4, -L + h_N]$, we have $|x - y| \le h_N$. Consequently, Equation (39) and $\rho h_N \le 1$ imply $\mu'(x) \le e\mu'(y)$. Integrating first over y gives

$$\frac{h_N}{4}\mu'(x) \le e \int_{-L+3h_N/4}^{-L+h_N} \mu'(y)\, dy.$$

Integrating this inequality over $x \in [-L, -L + 3h_N/4]$ and dividing by $h_N/4$, we obtain

$$\int_{-L}^{-L+3h_N/4} \mu'(x)\, dx \le 3e \int_{-L+3h_N/4}^{-L+h_N} \mu'(y)\, dy.$$

Therefore,

$$\mu\left(-L + \frac{3h_N}{4}\right) - \mu(-L) \le 3e \left[\mu(-L+h_N) - \mu\left(-L + \frac{3h_N}{4}\right) \right].$$

Each term in the preceding sum is at most Γ_μ. This also holds for the term with $h_0 = 2L$: although $2L$ is excluded from the supremum defining Γ_μ, continuity gives

$$\mu(L) - \mu(L/2) = \lim_{h \to 2L} \left[\mu(-L+h) - \mu\left(-L + \frac{3h}{4}\right) \right] \le \Gamma_\mu.$$

It follows that $\Delta_\mu \le (N+1+3e)\Gamma_\mu$. Moreover, the definition of N gives $N = \left\lceil \frac{\log\left(\max\{1, 2\rho L\}\right)}{\log(4/3)} \right\rceil$. Hence,

$$\Gamma_\mu \ge \frac{\Delta_\mu}{N + 1 + 3e}, \quad N = O\left(1 + \log(1 + \rho L)\right). \tag{40}$$

Since μ is strictly increasing, $\Gamma_\mu > 0$. By the definition of the supremum, we can choose $h \in (0, 2L)$ such that

$$\mu(-L+h) - \mu\left(-L + \frac{3h}{4}\right) \ge \frac{\Gamma_\mu}{2}.$$

We further set that $z_0 = -L + \frac{3h}{4}$, $z_1 = -L + h$, and $g = \mu(z_1) - \mu(z_0)$. Then $-L < z_0 < z_1 < L$ and $g \ge \Gamma_\mu/2$. We next construct parameters and matrices whose natural parameters are exactly z_0 and z_1.

Define $\alpha = \frac{|z_0|+h/4}{L}$, $\eta = \sqrt{\frac{h}{4\alpha L}}$, $c = \sqrt{1-\eta^2}$, and let $\xi = 1$ if $z_0 \geq 0$ and $\xi = -1$ otherwise. To verify that these quantities are well defined, observe that

$$\alpha = \begin{cases} 1 - h/(2L), & 0 < h \leq 4L/3, \\ h/L - 1, & 4L/3 < h < 2L. \end{cases}$$

Thus $0 < \alpha < 1$. Also, $\eta^2 = \frac{h/4}{|z_0|+h/4} \in (0, 1]$ and $c^2 = \frac{|z_0|}{|z_0|+h/4}$. In particular, $c^2 + \eta^2 = 1$, and

$$\xi \alpha L c^2 = z_0, \qquad \alpha L \eta^2 = \frac{h}{4} = z_1 - z_0. \tag{41}$$

These identities allow us to use the e_0 coordinate for a common reward level and coordinate i for the additional reward in instance i. Using the standard basis $e_0, \ldots, e_p$ of $\mathbb{R}^d$, define

$$\theta^{(i)} = S(ce_0 + \eta e_i),$$

for $i \in [p]$, and let $\theta^{(0)} = Sce_0$. All parameters are admissible, since $\left\|\theta^{(i)}\right\|_2 \leq S$ holds. The parameter $\theta^{(0)}$ will serve as the null environment.

We first consider $M \geq 1$. Take

$$\mathcal{A} = \{a_0, a_1, \ldots, a_p\}, \qquad a_0 = \alpha e_0, \qquad a_i = \alpha e_0 + \sqrt{1-\alpha^2}\, e_i, \quad i \in [p],$$

and initialize the memory at $s_1 = a_0^M$. Here, a_i^M denotes the memory state consisting of M copies of a_i. The action a_0 is distinct from every a_i, $i \in [p]$, because $\alpha < 1$. Define the memory matrix by

$$A(s)^\top = \begin{cases} R(\xi ce_0 + \eta e_j)e_0^\top, & s = a_j^M,\ j \in [p], \\ R\xi ce_0 e_0^\top, & \text{otherwise.} \end{cases} \tag{42}$$

Furthermore, it can be verified that $\|a_i\|_2 \leq 1$ holds for $i \in [p] \cup \{0\}$. The operator norm of a rank-one matrix $uv^\top$ is $\|u\|_2\|v\|_2$. Hence the two cases in Equation (42) have operator norms R and $Rc \leq R$, respectively. Moreover, every action in $\mathcal{A}$ satisfies $e_0^\top a = \alpha$. If $s = a_j^M$, then

$$\begin{aligned} \phi(s,a)^\top \theta^{(i)} &= \alpha R(\xi ce_0 + \eta e_j)^\top S(ce_0 + \eta e_i) \\ &= \alpha L\left(\xi c^2 + \eta^2 \mathbf{1}\{i = j\}\right) \\ &= z_0 + (z_1 - z_0)\mathbf{1}\{i = j\}. \end{aligned}$$

For every other state, the same inner product equals $\xi \alpha L c^2 = z_0$. Therefore, for all $a \in \mathcal{A}$,

$$\phi(s,a)^\top \theta^{(i)} = \begin{cases} z_1, & s = a_i^M, \\ z_0, & \text{otherwise.} \end{cases}$$

Under $\theta^{(0)}$, the e_i component is absent, so the natural parameter equals z_0 at every state–action pair. Thus, the only way to distinguish instance i from the null environment is to reach a_i^M.

For the first M rounds, the memory still contains at least one copy of the initial action a_0. None of the states a_i^M can therefore occur during these rounds. Every action sequence receives mean reward $\mu(z_0)$ for its first M rounds and at most $\mu(z_1)$ thereafter. Repeating a_i attains this upper bound in instance i, since its memory is a_i^M from round $M+1$ onward. Therefore,

$$\mathrm{OPT}_T(\theta^{(i)}) = M\mu(z_0) + (T-M)\mu(z_1) = T\mu(z_0) + (T-M)g.$$

We now compare the learner's trajectories across environments. Use the same internal random seed for the learner in every environment. Let $U_1, \ldots, U_T$ be independent uniform random variables, independent of that seed, and generate each reward observation as

$$y_t = F^{-1}_{\phi(s_t,a_t)^\top \theta}(U_t),$$

where F_z is the cumulative distribution function of the reward with natural parameter z, and F_z^{-1} is its inverse. This construction gives the correct reward distribution in each environment. It also ensures that whenever two environments have the same natural parameter at a given round, they receive the same observation at that round. On the trajectory under $\theta^{(0)}$, define

$$\tau_i = \inf\{t \in [T] : s_t = a_i^M\}, \quad \overline{\tau}_i = \min\{\tau_i, T+1\},$$

with $\inf \varnothing = \infty$. Until round τ_i, the trajectories under $\theta^{(i)}$ and $\theta^{(0)}$ coincide. Indeed, they start from the same memory and use the same learner seed. If their histories agree before some round $t < \tau_i$, they choose the same action, and both natural parameters equal z_0. Their observations are then identical under the coupling, so their histories also agree at the next round. This proves the claim by induction. Their states and selected actions still agree at round τ_i, but the observations at that round may differ; no agreement is needed thereafter.

Write $\mathrm{Reg}_T^{(i)}(\pi)$ for the regret on the trajectory under $\theta^{(i)}$. On this trajectory, the first $\overline{\tau}_i - 1$ mean rewards equal $\mu(z_0)$, and every remaining mean reward is at most $\mu(z_1)$. Thus,

$$\begin{aligned}\sum_{t=1}^{T} r_{\theta^{(i)}}(s_t^\pi, a_t^\pi) &\le (\overline{\tau}_i - 1)\mu(z_0) + (T - \overline{\tau}_i + 1)\mu(z_1) \\ &= T\mu(z_0) + (T - \overline{\tau}_i + 1)g.\end{aligned}$$

Subtracting the preceding upper bound from $\mathrm{OPT}_T(\theta^{(i)})$ yields

$$\mathrm{Reg}_T^{(i)}(\pi) \ge g(\overline{\tau}_i - M - 1).$$

The inequality holds for each realization of the common randomness, and hence also after taking expectations.

It remains to count how quickly the null trajectory can visit the p different states. First, $\tau_i \ge M + 1$ whenever τ_i is finite. Second, if $\tau_i < \tau_j < \infty$ for $i \ne j$, then $\tau_j - \tau_i \ge M$. Otherwise the two length-M action windows defining $s_{\tau_i} = a_i^M$ and $s_{\tau_j} = a_j^M$ would overlap, requiring an action in the overlap to equal both a_i and a_j, which is impossible. For each realization, order the truncated hitting times as $\overline{\tau}_{(1)} \le \cdots \le \overline{\tau}_{(p)}$. If $\overline{\tau}_{(j)} \le T$, the preceding separation argument gives

$$\overline{\tau}_{(j)} \ge M + 1 + (j-1)M = jM + 1.$$

If it equals $T + 1$, it corresponds to an unvisited state. In either case,

$$\overline{\tau}_{(j)} - (M+1) \ge \min\{(j-1)M, T - M\}.$$

Therefore,

$$\begin{aligned}\frac{1}{p}\sum_{i=1}^{p} \mathbb{E}\left[\mathrm{Reg}_T^{(i)}(\pi)\right] &\ge \frac{g}{p}\,\mathbb{E}\left[\sum_{j=1}^{p}\big(\overline{\tau}_{(j)} - M - 1\big)\right] \\ &\ge \frac{g}{p}\sum_{j=0}^{p-1}\min\{jM, T - M\} \\ &= \frac{g}{p}\sum_{j=0}^{p-1} jM = \frac{(p-1)Mg}{2}. \end{aligned} \tag{43}$$

The last equality uses $T \ge pM$, which implies $T - M \ge (p-1)M$.

For $M = 0$, keep the same parameter instances but instead take $A = RI_d$, $\mathcal{A} = \{a_1, \ldots, a_p\}$, and $a_i = \alpha(\xi c e_0 + \eta e_i)$. These actions are admissible because $\|a_i\|_2 = \alpha < 1$. For every $i, j \in [p]$, direct calculation gives

$$Ra_j^\top \theta^{(i)} = \alpha L\left(\xi c^2 + \eta^2 \mathbf{1}\{i = j\}\right) = z_0 + (z_1 - z_0)\mathbf{1}\{i = j\}.$$

Thus, action i has mean reward $\mu(z_1)$ in instance i, while every other action has mean reward $\mu(z_0)$. Under $\theta^{(0)}$, all actions again have mean $\mu(z_0)$. The optimal comparator always selects action i, so $\mathrm{OPT}_T(\theta^{(i)}) = T\mu(z_1)$.

Use the same coupling as above, and now let τ_i be the first round in which action i is selected on the null trajectory. Until that round, the observations in instance i and the null environment coincide. The first $\overline{\tau}_i - 1$ mean rewards in instance i therefore equal $\mu(z_0)$, which gives

$$\begin{aligned}\mathrm{Reg}_T^{(i)}(\pi) &\ge T\mu(z_1) - [(\overline{\tau}_i - 1)\mu(z_0) + (T - \overline{\tau}_i + 1)\mu(z_1)] \\ &= g(\overline{\tau}_i - 1).\end{aligned}$$

Only one action can be selected per round, so the j-th distinct action cannot be selected before round j. After ordering the truncated first-visit times, this gives $\overline{\tau}_{(j)} \geq j$ for $j \in [p]$; unvisited actions also satisfy this inequality because $T + 1 > p$. Consequently, for $T \geq p$,

$$\frac{1}{p}\sum_{i=1}^{p} \mathbb{E}\left[\mathrm{Reg}_T^{(i)}(\pi)\right] \geq \frac{g}{p}\sum_{j=1}^{p}(j-1) = \frac{(p-1)g}{2}. \tag{44}$$

In both cases, the supremum over admissible parameters is at least the average over the p constructed instances. Thus, Equation (43) and (44) give

$$\begin{aligned}\sup_{\theta^\star} \mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] &\geq \frac{(p-1)\max\{M,1\}g}{2} \\ &\geq \frac{(d-2)(M+1)\Gamma_\mu}{8} \\ &\geq \frac{(d-2)(M+1)\Delta_\mu}{8(N+1+3e)}.\end{aligned}$$

Here, the second inequality uses $p = d-1$, $\max\{M,1\} \geq (M+1)/2$, and $g \geq \Gamma_\mu/2$; the last inequality follows from Equation (40). Finally, $d-2 \geq d/3$ for $d \geq 3$ and $N = O(1+\log(1+\rho RS))$, proving the claimed lower bound. □

Lemma 7. *Let $d \geq 2$. If $T \geq \frac{v(d-1)^2}{U_\mu}\max\left\{\frac{1}{4R^2S^2}, \rho^2\right\}$, there exists an admissible action set such that, with $A(s) = RI_d$ for every s, every policy π satisfies*

$$\sup_{\theta^\star} \mathbb{E}\left[\mathrm{Reg}_T(\pi)\right] \geq \frac{3(d-1)}{16\sqrt{e}}\sqrt{vU_\mu T}.$$

Proof. Let $L = RS$ and $p = d-1$. We construct a family of instances whose natural parameters lie near a point where μ' attains U_μ. In this region, differences in the natural parameters translate into mean-reward differences proportional to U_μ. At the same time, we choose the instances sufficiently close to one another so that the learner cannot reliably distinguish their optimal actions within T rounds. We define $z_\star \in \arg\max_{|z|\leq L}\mu'(z)$, $\epsilon = \min\left\{\frac{L}{2}, \frac{1}{4\rho}\right\}$, and $\eta = \sqrt{\frac{\epsilon}{L}}$, where $1/(4\rho) = \infty$ when $\rho = 0$. Such a $z_\star$ exists because μ' is continuous on $[-L, L]$. Also, $0 < \eta^2 \leq 1/2$. Set $A(s) = RI_d$ for every s, and take

$$\mathcal{A} = \left\{a_u = \left(\sqrt{1-\eta^2}, \frac{\eta}{\sqrt{p}}u\right) : u \in \{-1,1\}^p\right\}.$$

The first coordinate is shared by all actions, while the remaining p coordinates can independently take either sign. Every action has unit norm, since

$$\|a_u\|_2^2 = 1 - \eta^2 + \frac{\eta^2}{p}\sum_{j=1}^{p} u_j^2 = 1.$$

For each $\omega \in \{-1,1\}^p$, define

$$\theta^{(\omega)} = \frac{1}{R}\left(z_\star\sqrt{1-\eta^2}, \nu\omega\right),$$

where $\nu = \frac{1}{4}\sqrt{\frac{vp}{U_\mu\eta^2 T}}$. All instances use the same action set and the same known memory matrix; only the unknown sign vector ω changes. We first verify that these parameters satisfy the norm bound. By the definition of ϵ,

$$\frac{1}{16\epsilon^2} = \max\left\{\frac{1}{4L^2}, \rho^2\right\}.$$

Thus, the condition on T is equivalent to $T \geq vp^2/(16U_\mu\epsilon^2)$, which gives

$$p\nu^2 = \frac{vp^2}{16U_\mu\eta^2 T} \leq \frac{\epsilon^2}{\eta^2} = L^2\eta^2.$$

Consequently,

$$R^2\|\theta^{(\omega)}\|_2^2 = z_\star^2(1-\eta^2) + p\nu^2 \le L^2(1-\eta^2) + L^2\eta^2 = L^2.$$

Therefore, $\|\theta^{(\omega)}\|_2 \le S$ for every ω. For action a_u under parameter $\theta^{(\omega)}$, the natural parameter is

$$Ra_u^\top \theta^{(\omega)} = (1-\eta^2)z_\star + \frac{\eta\nu}{\sqrt{p}} u^\top \omega.$$

The preceding norm bounds imply

$$|Ra_u^\top \theta^{(\omega)}| \le R\|a_u\|_2 \|\theta^{(\omega)}\|_2 \le L.$$

Moreover, $|u^\top \omega| \le p$ and $\nu\sqrt{p} \le L\eta$, so

$$\left| Ra_u^\top \theta^{(\omega)} - z_\star \right| \le \eta^2 |z_\star| + \eta\nu\sqrt{p} \le L\eta^2 + L\eta^2 = 2\epsilon.$$

Hence, all the constructed natural parameters lie in $[-L, L]$ and are within 2ϵ of $z_\star$. Since $\mu'(z_\star) = U_\mu$, the slope comparison in Equation (39) gives

$$\mu'(z) \ge e^{-\rho|z-z_\star|}\mu'(z_\star) \ge e^{-2\rho\epsilon} U_\mu \ge e^{-1/2} U_\mu$$

whenever $z \in [-L, L]$ and $|z - z_\star| \le 2\epsilon$. Together with the definition of U_μ, this yields

$$e^{-1/2} U_\mu \le \mu'(z) \le U_\mu \tag{45}$$

for every constructed natural parameter and every point between any two of them. These points all lie in $[-L, L]$, so the argument also applies when $z_\star$ is an endpoint.

We next bound how well the learner can distinguish two instances that differ in a single coordinate. Let P_z denote the reward distribution with natural parameter z. Taking the expectation of the log-likelihood ratio under P_z, and using $\mathbb{E}_{P_z}[y] = \mu(z)$, gives

$$\mathrm{KL}(P_z, P_{z'}) = \frac{\mu(z)(z - z') - b(z) + b(z')}{v}.$$

Since $b' = \mu$ and $b'' = \mu'$, Taylor's formula gives

$$b(z') - b(z) - \mu(z)(z' - z) = (z' - z)^2 \int_0^1 (1-\alpha)\mu'\big(z + \alpha(z' - z)\big)\, d\alpha \le \frac{U_\mu}{2}(z' - z)^2,$$

where we use the fact that the segment between z and z' lies in $[-L, L]$. Therefore,

$$\mathrm{KL}(P_z, P_{z'}) = \frac{b(z') - b(z) - \mu(z)(z' - z)}{v} \le \frac{U_\mu}{2v}(z - z')^2. \tag{46}$$

Fix any policy π. Write $\mathbb{P}_\omega$ for the distribution of its complete action–observation history under $\theta^{(\omega)}$, and let $\omega^{(j)}$ be obtained by reversing coordinate j of ω. For every available action a_u,

$$Ra_u^\top \theta^{(\omega)} - Ra_u^\top \theta^{(\omega^{(j)})} = \frac{2\eta\nu}{\sqrt{p}} u_j \omega_j.$$

Thus, the absolute difference between the two natural parameters is always $2\eta\nu/\sqrt{p}$, regardless of the selected action. By the divergence decomposition (Lattimore & Szepesvári, 2020, Lemma 15.1) and Equation (46),

$$\mathrm{KL}(\mathbb{P}_\omega, \mathbb{P}_{\omega^{(j)}}) = \sum_{t=1}^{T} \mathbb{E}_\omega \left[\mathrm{KL}\left(P_{Ra_{u_t}^\top \theta^{(\omega)}}, P_{Ra_{u_t}^\top \theta^{(\omega^{(j)})}} \right) \right] \le T \frac{U_\mu}{2v} \left(\frac{2\eta\nu}{\sqrt{p}} \right)^2 = \frac{2TU_\mu \eta^2 \nu^2}{pv} = \frac{1}{8},$$

where a_{u_t} denotes the selected action at round t. This decomposition compares the reward distributions at the same action conditional on a given history, and therefore remains valid for an adaptive policy. Pinsker's inequality now gives

$$\|\mathbb{P}_\omega - \mathbb{P}_{\omega^{(j)}}\|_{\mathrm{TV}} \le \sqrt{\frac{1}{2} \cdot \frac{1}{8}} = \frac{1}{4}.$$

We use this bound to show that the learner selects the wrong sign with a constant probability. Fix a round t and a coordinate j, and consider a pair $\omega, \omega^{(j)}$ with $\omega_j = 1$. An error in coordinate j occurs when $u_{t,j} = -1$ under ω, or when $u_{t,j} = 1$ under $\omega^{(j)}$. Consequently,

$$\begin{aligned}\mathbb{P}_\omega(u_{t,j} \neq \omega_j) + \mathbb{P}_{\omega^{(j)}}\left(u_{t,j} \neq \omega_j^{(j)}\right) &= \mathbb{P}_\omega(u_{t,j} = -1) + \mathbb{P}_{\omega^{(j)}}(u_{t,j} = 1) \\ &= 1 + \mathbb{P}_\omega(u_{t,j} = -1) - \mathbb{P}_{\omega^{(j)}}(u_{t,j} = -1) \\ &\geq 1 - \|\mathbb{P}_\omega - \mathbb{P}_{\omega^{(j)}}\|_{\mathrm{TV}} \geq \frac{3}{4}.\end{aligned}$$

The inequality follows because the difference in the probability of the same event under two distributions is bounded by their total variation distance. For each coordinate j, the 2^p sign vectors form 2^{p-1} disjoint pairs of this type. Summing over these pairs and dividing by 2^p gives that, for every round t and every coordinate j,

$$\frac{1}{2^p} \sum_{\omega \in \{-1,1\}^p} \mathbb{P}_\omega(u_{t,j} \neq \omega_j) \geq \frac{2^{p-1}}{2^p} \frac{3}{4} = \frac{3}{8}. \tag{47}$$

Finally, we relate these coordinate errors to regret. Since μ is increasing, an optimal action maximizes $u^\top \omega$, which is achieved by $u = \omega$. Thus, a_ω is optimal under $\theta^{(\omega)}$. For any selected action a_{u_t},

$$R(a_\omega - a_{u_t})^\top \theta^{(\omega)} = \frac{\eta\nu}{\sqrt{p}} \sum_{j=1}^{p} (1 - u_{t,j}\omega_j) = \frac{2\eta\nu}{\sqrt{p}} \sum_{j=1}^{p} \mathbf{1}\{u_{t,j} \neq \omega_j\}.$$

In particular, each incorrect coordinate contributes $2\eta\nu/\sqrt{p}$ to the difference in natural parameters. The slope lower bound in Equation (45) holds throughout the interval between these parameters. Integrating μ' over this interval therefore gives

$$\mu(Ra_\omega^\top \theta^{(\omega)}) - \mu(Ra_{u_t}^\top \theta^{(\omega)}) \geq \frac{U_\mu}{\sqrt{e}} R(a_\omega - a_{u_t})^\top \theta^{(\omega)} = \frac{2U_\mu \eta\nu}{\sqrt{ep}} \sum_{j=1}^{p} \mathbf{1}\{u_{t,j} \neq \omega_j\}.$$

Because $A(s) = RI_d$, the memory state does not affect rewards, and the optimal comparator selects a_ω in every round. Summing the preceding inequality over T rounds, taking expectations, and averaging over the parameter instances yields

$$\begin{aligned}\frac{1}{2^p} \sum_{\omega \in \{-1,1\}^p} \mathbb{E}_\omega\left[\mathrm{Reg}_T(\pi)\right] &\geq \frac{2U_\mu \eta\nu}{\sqrt{ep}} \sum_{t=1}^{T} \sum_{j=1}^{p} \frac{1}{2^p} \sum_{\omega \in \{-1,1\}^p} \mathbb{P}_\omega(u_{t,j} \neq \omega_j) \\ &\geq \frac{2U_\mu \eta\nu}{\sqrt{ep}} \frac{3pT}{8} = \frac{3p}{16\sqrt{e}} \sqrt{vU_\mu T}.\end{aligned}$$

The second inequality follows from Equation (47), and the last equality uses the definition of ν. Since all the constructed parameters satisfy $\|\theta^{(\omega)}\|_2 \leq S$, the supremum over admissible parameters is at least this average. Substituting $p = d - 1$ proves the lemma. □