MULTISCALE BUFFERED INTERLEAVING: CONSTANT-GAP PLANNING AND ONLINE LEARNING FOR RECOVERING REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Motivated by the need to keep large collections of AI-agent memories up to date, we study a recovering-reward bandit problem. In each round, at most K of n arms can be pulled, and each arm's expected reward is a bounded, nondecreasing function of the time since its last pull. We develop Multiscale Buffered Interleaving (MBI), which aggregates compatible periodic plans without reward loss and coordinates the remaining requests through multiscale buffering. For the offline problem with known reward curves, MBI achieves a long-run average reward within L of the renewal upper bound, where L bounds the reward of a single pull. This system-wide loss bound is independent of n, K, and the recovery horizon, improving previous $O(L\sqrt{K})$ guarantees, and a constant-order loss is necessary. For the online problem with unknown reward curves, the proposed policy achieves $\widetilde{O}(nL\sqrt{T})$ regret against our offline planning benchmark. A lower bound on growing instance families matches this learning order up to logarithmic factors. Experiments show lower losses than previous purely periodic and randomize-then-interleave methods in both offline and online settings.

1 INTRODUCTION

Recent advances in large language models have fueled growing interest in AI agents and their use across a wide range of applications (Wang et al., 2024). Among these applications, assistants that interact with users over time can use memory to draw on earlier conversations, preferences, and experiences. Systems such as MemGPT and Mem0 support this use of memory by retaining relevant information across conversations and retrieving it to inform later responses (Packer et al., 2023; Chhikara et al., 2025). As these interactions accumulate, the stored information can reflect facts and preferences recorded at different times. Some may remain relevant, while others may be superseded by new information. Keeping this information useful therefore involves updating existing memories (Rasmussen et al., 2025; Xu et al., 2026).

Maintaining this growing body of memory raises a practical question: how frequently should different memories be updated under a limited computational budget? Mem0 and Zep use periodic refreshes in memory maintenance (Chhikara et al., 2025; Rasmussen et al., 2025). LightMem and LycheeMemory combine information from multiple interactions to reduce repeated LLM calls (Fang et al., 2026; Li et al., 2026). These designs motivate a tradeoff: waiting can allow each update to incorporate more information, but also means updating less frequently. When many memories share the same resources, the resulting scheduling problem is to decide which memories to update now and which to leave for later.

The scheduling problem also arises beyond memory maintenance. In e-commerce promotions and product recommendations, customer interest can recover after a period without exposure to the same product (Simchi-Levi et al., 2021; Pike-Burke & Grunewalder, 2019). Similar recovery effects appear in recurring notifications, where messages not shown recently can regain their effectiveness (Yancey & Settles, 2020). Web crawling and spaced repetition have also motivated recovering-bandit models (Kleinberg & Immorlica, 2018). These settings motivate a common modeling idea: an action's expected reward can recover with the time since it was last taken. We study this mecha-

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.

nism in a recovering-reward bandit with n arms, of which at most K can be pulled each round. Each arm's expected reward is bounded and nondecreasing in the time since its last pull.

Previous work has established asymptotically optimal planning guarantees for recovering rewards. For concave recovery with one pull per round, a polynomial-time approximation scheme uses interleaved rounding to coordinate randomly shifted periodic requests and control fluctuations in waiting intervals (Kleinberg & Immorlica, 2018). For bounded, nondecreasing recovery curves with multiple pulls per round, purely periodic policies (PPP) achieve a long-run approximation ratio of $1 - O(1/\sqrt{K})$ (Simchi-Levi et al., 2021). Randomize-Then-Interleave (RTI) provides an approximation ratio of $1 - K^K/(e^K K!)$ in expectation when each reward curve becomes constant after a finite wait (Papadigenopoulos et al., 2022). These ratios do not ensure a capacity-independent system-wide loss bound, which is especially important when K is large.

The planning guarantees above assume known reward curves, but applications may require scheduling before these curves have been learned (Pike-Burke & Grunewalder, 2019; Yancey & Settles, 2020). For memory maintenance, this means choosing updates while learning how their value depends on the waiting period. The same actions thus generate both rewards and information for future decisions (Kleinberg & Immorlica, 2018; Simchi-Levi et al., 2021). Recovery introduces a further tradeoff: pulling sooner provides feedback earlier but resets recovery, whereas waiting may improve reward per pull but slows the collection of observations (Simchi-Levi et al., 2021; Papadigenopoulos et al., 2022). An online policy balances these competing needs under the shared capacity constraint.

Can we achieve a capacity-independent planning loss and learn to attain the same benchmark?

1.1 Our Contributions

We summarize our contributions as follows.

1. **A constant-gap offline planning framework.** We propose Multiscale Buffered Interleaving (MBI), which aggregates compatible periodic plans without loss and coordinates the remaining pulls through multiscale buffering. For known, bounded, nondecreasing reward curves, MBI achieves a long-run average reward within L of the renewal upper bound, where L bounds the reward of a single pull (Theorem 1). This system-wide loss bound is independent of n, K, and the recovery horizon, strengthening previous asymptotically optimal relative guarantees (Simchi-Levi et al., 2021; Papadigenopoulos et al., 2022) to a capacity-independent absolute guarantee.
2. **Necessity of a constant planning loss.** We prove that a constant-order planning loss is unavoidable. For every capacity K, we construct an instance whose optimal long-run average reward is exactly $5L/36$ below the renewal upper bound (Theorem 2). Together with Theorem 1, this establishes the optimal $\Theta(L)$ order of the system-wide planning loss guarantee.
3. **An online learning-while-planning algorithm.** We extend MBI to unknown reward curves by combining optimistic planning with block execution. The proposed online policy achieves $\widetilde{O}(nL\sqrt{T})$ expected regret against the same constant-gap planning benchmark over T rounds (Theorem 3). A learning lower bound shows that this order cannot be improved uniformly over the model class by more than logarithmic factors (Theorem 4).

1.2 Literature Review

Recovering-reward planning and learning. Kleinberg & Immorlica (2018) introduced recharging bandits and studied concave recovery with one pull per round. They combine interleaved rounding with partial enumeration to obtain a polynomial-time approximation scheme, and extend the planner to online learning through upper confidence bounds. Simchi-Levi et al. (2021) extend planning to heterogeneous monotone recovery with multiple pulls through purely periodic policies (PPP). Their capacity-dependent approximation guarantees are near-optimal within the PPP class, up to logarithmic factors (Simchi-Levi et al., 2025). Randomize-Then-Interleave (RTI) improves the approximation factors through LP rounding and interleaving for curves with finite recovery horizons (Papadigenopoulos et al., 2022). Foussoul et al. (2023) obtain constant-factor guarantees for monotone last-switch-dependent rewards with finite saturation times.

For unknown rewards, Simchi-Levi et al. (2021) also use UCB-based planning, while RTI and the monotone last-switch extension use explore-then-commit learning. Other approaches use Gaussian-process priors (Pike-Burke & Grunewalder, 2019), shared recovery shapes (Cella & Cesa-Bianchi, 2020), or discounted finite-age models (Tanimoto & Fukumizu, 2026). Feature-based action-history models obtain learning guarantees using an exact block-optimization oracle (Ann et al., 2026).

Related models cover general last-switch dependence (Laforgue et al., 2022), blocking under stochastic or adversarial rewards (Basu et al., 2019; Atsidakou et al., 2021; Cesa Bianchi et al., 2026), arriving requests and resource matching (Sumida, 2024; Freund & Hess, 2026), and costly introduction of new interventions (Kalvit & Singhvi, 2025). Our contribution is a constructive planner whose additive loss relative to the renewal upper bound is independent of n and K, together with an online policy that attains the same guarantee up to a sublinear learning loss.

Scheduling and resource allocation. Periodic scheduling studies recurring service under shared capacity (Anily & Bramel, 2000; Bar-Noy et al., 2002). LP and Lagrangian relaxations guide stochastic resource-allocation policies through tractable upper bounds (Bertsimas & Niño-Mora, 2000; Adelman & Mersereau, 2008). Brown & Zhang (2025) establish constant-order performance gaps for fluid-budget balancing under nondegeneracy and a fixed number of types, while Zhang et al. (2026) obtain asymptotic optimality for fully heterogeneous average-reward systems under mixing conditions. Fair scheduling and dependent rounding coordinate service shares under capacity constraints (Baruah et al., 1993; Gandhi et al., 2006). Our analysis links capacity-feasible slot allocation to the waiting intervals that determine recovering rewards.

2 Model

We study a recovering-reward multi-armed bandit problem with n arms. In each round $t = 1, 2, \ldots$, a set A_t of arms is pulled subject to the capacity constraint

$$A_t \subseteq [n], \qquad |A_t| \le K, \qquad 1 \le K \le n. \tag{1}$$

Let $d_i(t)$ denote the number of rounds since arm i was last pulled, measured before the action in round t. A pull in round t gives $d_i(t+1) = 1$. Initially, $d_i(1) = a_i$, where $a = (a_1, \ldots, a_n)$ is a vector of positive integers.

Pulling arm i in round t yields a random reward $Y_{i,t}$ with mean $R_i(d_i(t))$. Each reward curve R_i is nondecreasing and takes values in $[0, L]$, where $L > 0$ is known. We write $R = (R_1, \ldots, R_n)$ for the collection of reward curves. Following Simchi-Levi et al. (2021), we do not require concavity, unlike the earlier setting of Kleinberg & Immorlica (2018). We also allow curves that never become constant after a finite wait. Finite-time recovery is assumed by Papadigenopoulos et al. (2022) and used to compute the offline relaxation in Simchi-Levi et al. (2021).

For a horizon T, we seek a policy π maximizing the expected cumulative reward $\mathbb{E}_\pi[S_T]$, where $S_T = \sum_{t=1}^{T} \sum_{i \in A_t} Y_{i,t}$. For any fixed initial ages, we define the optimal long-run average reward as

$$\mathrm{OPT}(R) = \sup_{\pi} \liminf_{T \to \infty} \frac{\mathbb{E}_\pi[S_T]}{T}.$$

In the offline problem, the reward curves are known in advance. In the online problem, they are unknown, and only the realized rewards of the pulled arms are observed. These observations lie in $[0, L]$ and satisfy

$$\mathbb{E}[Y_{i,t} \mid \mathcal{F}_{t-1}, A_t] = R_i(d_i(t)) \qquad (i \in A_t), \tag{2}$$

where $\mathcal{F}_{t-1}$ denotes the history through round $t-1$, including the policy's internal randomization.

3 Offline Planning with a Constant Gap

In this section, we consider the offline problem defined in Section 2. With known reward curves, offline planning requires deciding how often to pull each arm and how to fit these pulls within the per-round capacity.

3.1 PLANNING WITH MBI

We first choose the frequencies, enforcing capacity only on average. Similar to Simchi-Levi et al. (2021) and Papadigenopoulos et al. (2022), we formulate these choices through a renewal relaxation.

Let x_{ip} denote the long-run fraction of arm i's timeline covered by waiting intervals of length p. Pulling arm i every p rounds yields average reward $R_i(p)/p$ and contributes $1/p$ pulls per round. Each arm's time fractions sum to at most one, with any unallocated share corresponding to idling, and the total average number of pulls per round cannot exceed K. We have

$$\begin{aligned} \mathrm{UB}(R) = \sup_{x \geq 0} \quad & \sum_{i=1}^{n} \sum_{p \geq 1} \frac{R_i(p)}{p} x_{ip} \\ \text{subject to} \quad & \sum_{p \geq 1} x_{ip} \leq 1 \quad (i \in [n]), \qquad \sum_{i=1}^{n} \sum_{p \geq 1} \frac{x_{ip}}{p} \leq K. \end{aligned} \tag{3}$$

The value $\mathrm{UB}(R)$ upper-bounds the optimal long-run average reward and serves as our benchmark. Our construction exploits the following structural property of the LP.

Lemma 1. *When only finitely many periods are included in the LP (3), it admits an optimal basic solution x^* in which all but at most one arm satisfy $x^*_{ip} \in \{0, 1\}$ for every p. The exceptional arm, if any, mixes at most two choices, counting idling as a choice.*

The LP determines how often to pull the arms, but their pulls may still fall in the same round and exceed capacity. We propose MBI to turn these pull frequencies into a schedule that respects the per-round capacity. Figure 1 illustrates the construction, which we explain step by step below.

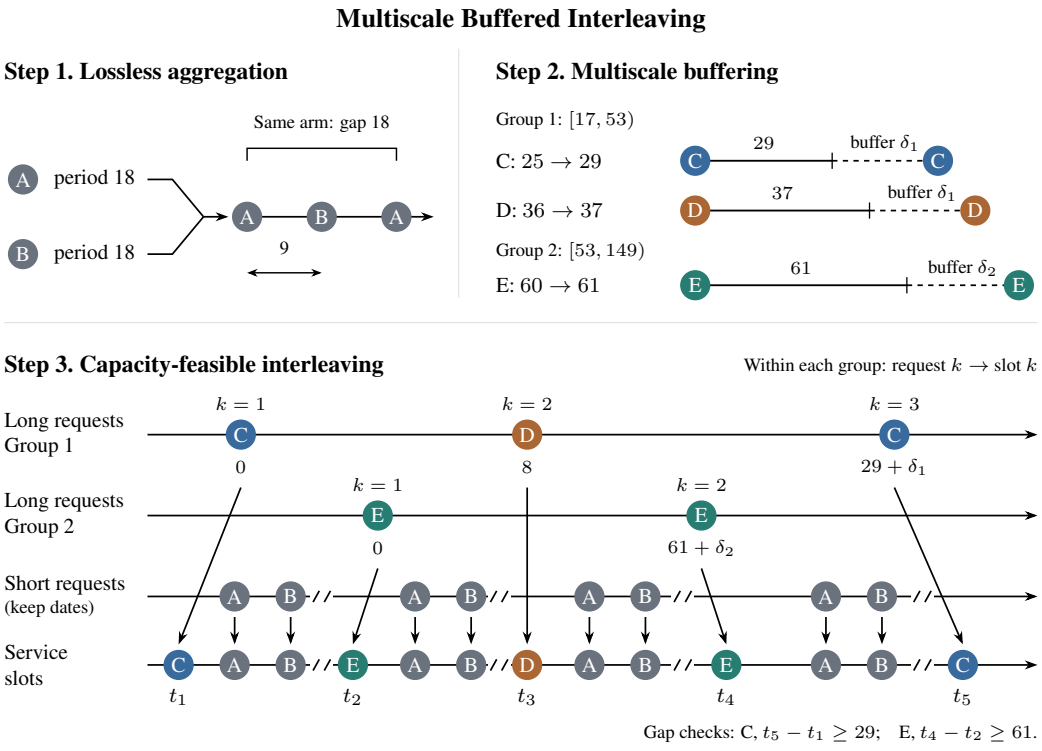


Figure 1: The three steps of MBI, illustrated with LP-selected periods $A = B = 18$, $C = 25$, $D = 36$, and $E = 60$. Aggregation preserves each arm's wait. We add buffers δ_1, δ_2 between requests in the long-period groups to protect their waiting intervals during order-preserving interleaving. Short requests keep their dates. Axis breaks omit intermediate rounds.

Step 1. Lossless aggregation. We first exploit divisibility to combine compatible periodic plans without loss. Cycling through m streams of period p, with $m \mid p$, produces a stream of period p/m while preserving each underlying arm's nominal waiting interval and reward rate. Recursively merging groups of $\mathrm{spf}(p)$ streams in decreasing period order leaves $r_p < \mathrm{spf}(p)$ residual streams at each $p \geq 2$, where $\mathrm{spf}(p)$ is the smallest prime factor of p. We reserve one slot in every round for each period-one stream. Only the residual streams require further coordination.

Step 2. Multiscale buffering. We round the remaining long periods upward to a smaller set of nearby values. We group streams with similar periods and add extra time between planned pulls (requests) to protect their waits during interleaving. Longer periods can accommodate larger buffers at a smaller frequency cost. These buffers balance two losses: serving less often and discarding requests whose waits become too short.

Step 3. Capacity-feasible interleaving. After reserving the complete one-slot schedules, MBI divides the remaining daily capacity among the request groups. Short-period requests keep their coordinated dates, with excess requests discarded. Long-period requests keep their order: within each group, the k-th request takes the k-th assigned service slot. The buffers from Step 2 help preserve the required waits when service dates shift. Long-period requests that still fail the waiting-time check are removed without moving later requests.

3.2 Planning guarantees

Table 1 compares planning guarantees by their long-run loss relative to $\mathrm{UB}(R)$. Previous guarantees for K pulls per round are multiplicative. Purely periodic policies (PPP) (Simchi-Levi et al., 2021) schedules each arm strictly periodically, and its loss fraction $O(1/\sqrt{K})$ is within a logarithmic factor of the best that purely periodic policies can guarantee (Simchi-Levi et al., 2025). Randomize-Then-Interleave (RTI) (Papadigenopoulos et al., 2022) assumes that every curve becomes constant after a known time, gives arms independent random offsets, and skips requests beyond capacity. When $\mathrm{UB}(R) = \Theta(KL)$, both guarantees allow a loss of order $L\sqrt{K}$ per round. For concave recovery with one pull per round, Kleinberg & Immorlica (2018) give an approximation scheme that serves randomly shifted periodic requests in order.

Table 1: Upper and lower bounds on the long-run planning loss relative to $\mathrm{UB}(R)$. A dash indicates that the corresponding paper does not establish a matching lower bound.

Method	Loss upper bound	Loss lower bound	Scope of the lower bound
PPP (Simchi-Levi et al. (2025))	$O\left(\frac{\mathrm{UB}(R)}{\sqrt{K}}\right)$	$\Omega\left(\frac{\mathrm{UB}(R)}{\sqrt{K \log K}}\right)$	Purely periodic policies
RTI (Papadigenopoulos et al. (2022))	$\frac{K^K}{e^K K!}\,\mathrm{UB}(R)$	—	—
MBI (this work)	L	$\frac{5L}{36}$	All feasible policies

MBI extends such order-preserving interleaving to K slots through exact per-round quotas, without concavity or a finite recovery horizon. Its loss is at most L, and no policy can guarantee a loss below $5L/36$ (Theorems 1 and 2). For a calendar C of length h and initial ages a, we denote $J_R(C; a) = \sum_{t=1}^{h} \sum_{i \in C_t} R_i(d_i(t))$ as its cumulative expected reward.

Theorem 1. *For any initial ages a and length $h \geq 1$, MBI returns a feasible calendar C satisfying*

$$J_R(C; a) \geq h\{\mathrm{UB}(R) - L\} - nL. \tag{4}$$

Consequently, MBI achieves a long-run average reward of at least $\mathrm{UB}(R) - L$.

Proof sketch. Dependent rounding gives each group of long-period streams an integer number of slots in every round, and all groups together fill the capacity exactly (Lemma A.3). Within a group, the kth request takes the kth slot, so a mismatch between requests and slots only shifts service dates. Long-period requests therefore lose reward in two main ways: buffers lower their pull frequency, and a request whose wait still falls short is deleted. For a group whose periods are at least P and

take d distinct values, let F be the expected number of its requests during the extra time added to each wait. Per round,

$$\text{frequency loss} \le L\,\frac{F}{P}, \qquad \text{deletion loss} \le L\,\frac{d+3}{4eF}\exp\Big(-\frac{2F^2}{d+3}\Big) \tag{5}$$

(Lemmas A.4 and A.5). The deletion bound follows from Hoeffding's inequality: equal-period streams share one evenly spaced pool, so the fluctuation a buffer must absorb is a sum of at most $d-1$ independent terms of range one, plus a bounded quota error. Neither bound depends on the number of arms. Groups cover period ranges $[P, 3P)$ on geometric scales, so $d = O(P)$; choosing F slightly above $\sqrt{d}$ makes both losses decay geometrically across scales, and their total is bounded independently of n and K. Periods at most 16 are not buffered: after lossless aggregation (Lemma A.1), their requests form at most seven independent packages, whose overflow is $O(1)$. With sharper estimates for periods below 449, a certificate bounds the per-round loss, including period rounding and LP truncation, by $0.98L$ (Appendix A.5). The unscored first pull of each arm gives the $-nL$ term.

The following theorem shows that a constant-order gap to the renewal upper bound is unavoidable.

Theorem 2. *For every $K \ge 1$, there exists an instance for which*

$$\mathrm{UB}(R) - \mathrm{OPT}(R) = 5L/36.$$

Together, Theorems 1 and 2 establish the optimal order of the additive planning loss guarantee. We next study how to learn to attain the same benchmark when the reward curves are unknown.

4 Online Learning

In this section, we turn to address the online counterpart of the offline planning problem. When reward curves are unknown, deciding how long to wait also determines what can be learned. Short waits provide rewards and feedback more often. Longer waits may increase the reward of a pull, but postpone both that reward and the opportunity to observe it. The learner have to resolve this tradeoff while approaching the offline planning guarantee.

4.1 Learning with MBI

We combine MBI with Kullback–Leibler upper confidence bounds (KL-UCB; Garivier & Cappé, 2011) to obtain our online policy MBI-KL.

Step 1: Construct optimistic reward curves. For arm i at wait d, let $N_i(d)$ count retained observations and $\widehat{\mu}_i(d)$ be their mean divided by L. With confidence parameter β_j, set $U_i(d) = L$ when $N_i(d) = 0$; otherwise, $U_i(d) = L \sup\{q \in [\widehat{\mu}_i(d), 1] : N_i(d)\,\mathrm{kl}(\widehat{\mu}_i(d), q) \le \beta_j\}$. Here $\mathrm{kl}(x, y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$. We then replace $U_i(d)$ with $\min_{q \ge d} U_i(q)$ to give MBI nondecreasing reward curves. We call this procedure KL-UCB.

Step 2: Execute MBI in blocks. We divide time into epochs, with epoch j consisting of rounds $\{2^j, \dots, 2^{j+1} - 1\}$. Epoch j sets the learning range to waits $d \le 2^j$, allowing progressively longer waits to be learned without a known recovery horizon.

Each epoch is further divided into blocks of h_j rounds, shortened at the epoch boundary. Each block executes one MBI calendar C, constructed from U and the current ages a. The block length balances calendar startup costs against delayed use of new feedback.

Step 3: Update the reward estimates. After each block, we add only the new observations with actual waits $d \le 2^j$ to the retained history. This history is used to update U for the next MBI plan. Retained observations and current ages carry over across epochs.

4.2 Learning guarantees

We prove that Algorithm 1 achieves the same theoretical guarantee as in the offline problem, up to an additional loss from learning.

Algorithm 1 MBI-KL

Input: Number of arms n, capacity K, reward bound L, initial ages a, confidence parameters (β_j), block lengths (h_j).

$t \leftarrow 1$ and initialize empty observation history.
for $j = 0, 1, \ldots$ **do**
  while $t < 2^{j+1}$ **do**
    $h \leftarrow \min\{h_j, 2^{j+1} - t\}$.
    $U \leftarrow$ KL-UCB(retained observations; β_j).
    $C \leftarrow \mathrm{MBI}(U, K, L, a, h)$.
    Execute C and advance the current ages a.
    Retain new observations with actual wait $d \le 2^j$.
    $t \leftarrow t + h$.
  end while
end for

Theorem 3 (Online learning guarantee)**.** *For all $T \ge 1$, Algorithm 1 satisfies*

$$\frac{\mathbb{E}S_T}{T} \ge \mathrm{UB}(R) - L - \widetilde{O}\left(\frac{nL}{\sqrt{T}}\right). \tag{6}$$

For rational inputs and observations, expected computation is polynomial in their encoded length and the number of output rounds.

For a fixed system, $\widetilde{O}(nL/\sqrt{T}) \to 0$ as $T \to \infty$. Thus the additional learning term vanishes, recovering the offline guarantee.

The next theorem gives a matching lower bound up to logarithmic factors.

Theorem 4 (Learning lower bound)**.** *There is an absolute constant $c > 0$ such that, for any online policy, there is a growing family of instances and horizons $T \to \infty$ for which*

$$\frac{\mathbb{E}S_T}{T} \le \mathrm{UB}(R) - L - c\,\frac{nL}{\sqrt{T}}. \tag{7}$$

Together, Theorems 3 and 4 show that the learning order cannot be improved uniformly over the model by more than logarithmic factors.

5 Experiments

In this section, we use synthetic experiments to demonstrate the performance of our algorithms. Our experiments have two parts: the offline experiments evaluate how well the algorithms schedule pulls when the reward curves are known. The online experiments evaluate their performance when these curves are unknown and must be learned from observed rewards.

Both studies evaluate the algorithms under four recovery settings: *Continuous*, with several smooth recovery stages per arm, *Staircase*, with discrete recovery steps, *Hybrid*, combining smooth and discrete stages within each arm, and *Mix*, combining the three arm types in one instance. Stage lengths and reward increments vary across arms.

The central experiments use $n = 2048$, $K = D = 128$, and three paired seeds per family, where D denotes the largest saturation age. Offline and online methods share the same reward tables.

We evaluate both settings using the full-horizon loss to the renewal upper bound:

$$\mathrm{Loss}(T) = \mathrm{UB}(R) - \frac{1}{T}\sum_{t=1}^{T}\sum_{i \in A_t} R_i(d_i(t)), \tag{8}$$

where $R_i(d_i(t))$ is the conditional mean reward of the executed pull. Appendix E provides the generators, implementation details, and supporting results.

5.1 Experiments on Offline planning

With known reward curves, we compare MBI with Randomize-Then-Interleave (RTI) of Papadigenopoulos et al. (2022) and the purely periodic planner (PPP) of Simchi-Levi et al. (2025). The central and capacity-scaling comparisons run for $T = 32768$ rounds. Figure 2(a) shows the central comparison. MBI has the smallest loss in each family, with means between 0.377 and 0.419, versus 0.702–1.084 for RTI and 1.002–1.064 for PPP.

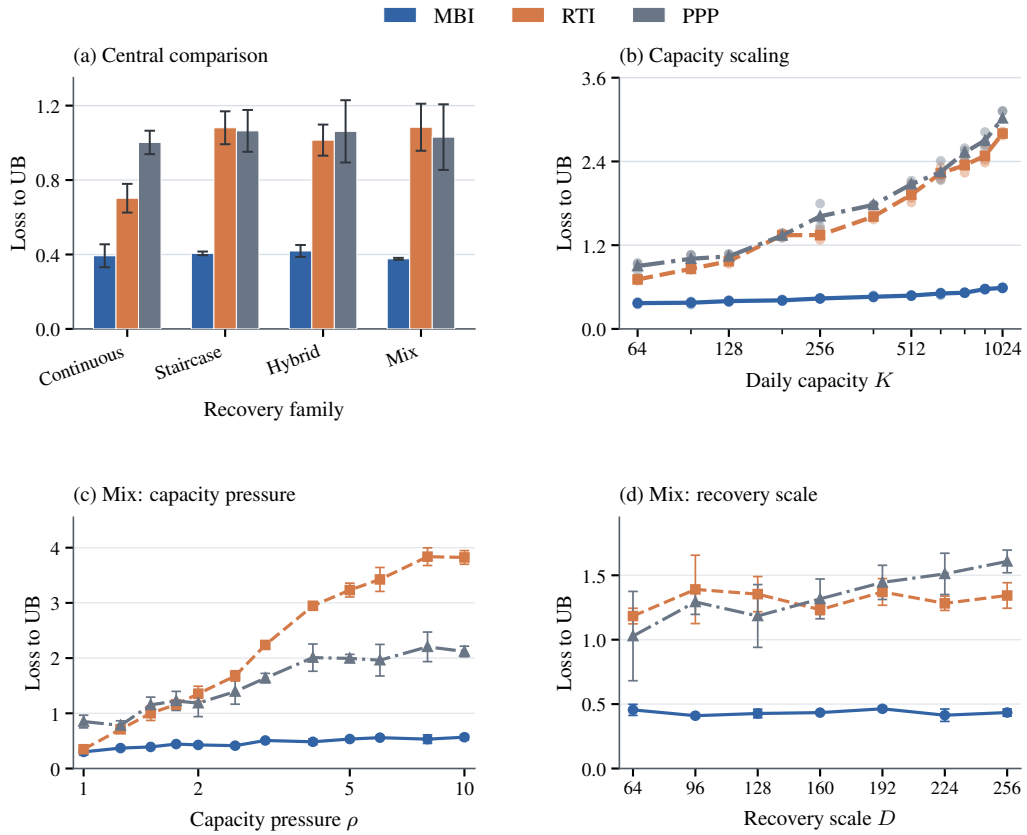


Figure 2: Full-horizon offline loss to the renewal upper bound. Lower is better. (a) Central setting: bars show means and error bars show sample standard deviations over three paired seeds. (b) Capacity scaling with $n = 16K$ and $D = 128$: each seed is averaged equally over the four families; lines show three-seed means and light points show individual seeds. (c) Mix capacity pressure at $K = D = 128$. (d) Mix recovery scale at $K = 128$ and $\rho \approx 2$, adjusting n to maintain pressure. Panels (a)–(c) use $T = 32768$; panel (d) uses $T = 256D$. In (c)–(d), lines show three-seed means and error bars show sample standard deviations.

We increase K from 64 to 1024 with $n = 16K$, while keeping $D = 128$. In Figure 2(b), MBI's mean full-horizon loss increases from 0.369 to 0.588, compared with 0.709 to 2.800 for RTI and 0.901 to 3.017 for PPP. MBI maintains a small absolute loss as the system scales, with substantially slower growth than either baseline, consistent with its capacity-independent long-run guarantee. After excluding the first 128 rounds, MBI's loss remains approximately constant across capacities (Figure 5 in Appendix E.3).

Let $p_i^\star$ maximize $R_i(p)/p$. We define capacity pressure as $\rho = (\sum_i 1/p_i^\star)/K$, the total independently preferred pulling frequency divided by capacity. Across pressure targets $\rho = 1$ to 10, MBI maintains lower full-horizon loss than both baselines, as illustrated on Mix in Figure 2(c). At matched pressure $\rho \approx 2$, its loss also remains stable and below both baselines as D varies from 64 to 256 (Figure 2(d)). Appendix E.3 presents both scans on all four families.

5.2 Experiments on Online learning

With unknown reward curves, we compare MBI-KL with PPP (Simchi-Levi et al., 2025) and an adaptation of RTI. The RTI learner of Papadigenopoulos et al. (2022) collects 12 observations per arm–age pair under our parameter choices, requiring 1,585,168 rounds of batchwise exploration, far exceeding our horizon $T = 16{,}384$. We therefore collect only one observation per arm–age pair using pipelined exploration, fit monotone reward curves, and apply RTI's scheduling rule. We call this adaptation RTI-A.

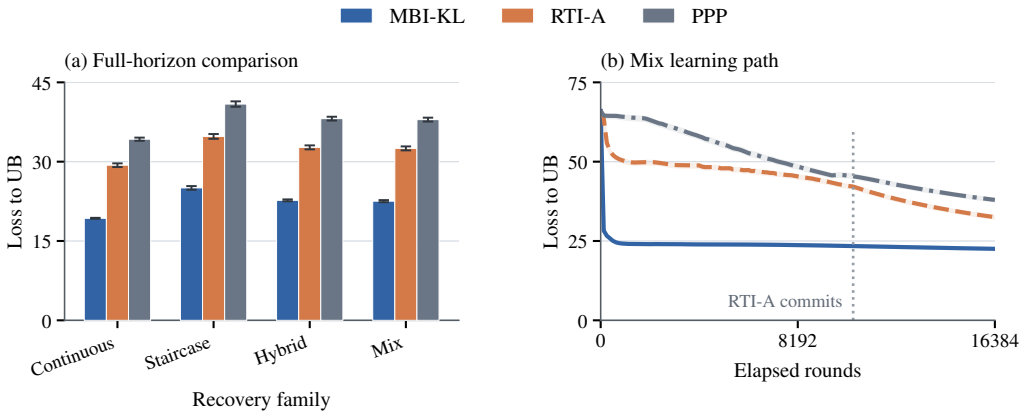


Figure 3: Online loss to the renewal upper bound in the central setting at $T = 16384$. Lower is better. (a) Full-horizon comparison: bars show means and error bars show sample standard deviations over three paired seeds. (b) Mix learning paths, with every point averaging from round one: lines show three-seed means and shading shows the seed range. The vertical line marks RTI-A's commitment at round 10494.

MBI-KL has the lowest full-horizon loss on all four families and all 12 paired workloads. The learning paths in Figure 3 show why the full process matters. RTI-A improves after exploration and attains a lower final-quarter loss on Mix (15.33, versus MBI-KL's 20.76), but its earlier cost leaves a larger all-round loss. MBI-KL earns more reward over the full learning horizon in this comparison.

For load pressure, we compare MBI-KL and RTI-A on Mix at $K = D = 128$, $T = 32768$, and $\rho \in \{1, 2, 3, 5, 10\}$, adjusting population size to match each target. MBI-KL has lower full-horizon loss on all 15 paired workloads; its reduction in three-seed mean loss relative to RTI-A ranges from 14.0% to 56.7%. RTI-A's exploration occupies approximately 29%–87% of the horizon as pressure increases. Over the common window after RTI-A's commitment, RTI-A has lower loss at $\rho = 1, 2, 3, 5$, while MBI-KL has lower loss at $\rho = 10$. These results connect the full-horizon advantage to the cost of exploration under load (Figure 9 and Table 4).

We also examine learning horizon, feedback noise, and replanning frequency on Mix. At $T = 32768$, MBI-KL's loss decreases from 22.62 to 20.97, while PPP improves from 38.16 to 30.78. More frequent replanning gives a modest further improvement but roughly doubles planning calls. Lower-noise feedback preserves the all-round ranking, although RTI-A benefits more after exploration. Appendix E.4 contains all online figures and tables, including these sensitivity analyses.

6 Conclusion

We introduced Multiscale Buffered Interleaving (MBI), achieving a system-wide long-run planning loss of at most L relative to the renewal upper bound, independently of population size, capacity, and recovery horizon. A lower bound establishes the optimal $\Theta(L)$ order. MBI-KL attains $\widetilde{O}(nL\sqrt{T})$ regret against the constant-gap benchmark, matched up to logarithmic factors on growing instance families. Experiments show small absolute planning losses as capacity grows and lower full-horizon online losses on the tested workloads, including under varying load pressure.

AI USE STATEMENT

We used AI tools to assist with polishing the writing and refining the figures. All research ideas, technical results, analyses, and conclusions were developed and verified by the authors.

REPRODUCIBILITY STATEMENT

Appendices A–D give the proofs of Theorems 1–4 and algorithm details. Appendix E describes the experimental setup, baseline implementations, and supporting results.

ETHICS STATEMENT

This work develops planning and learning methods for recovering-reward problems and evaluates them using synthetic data. The experiments involve no human subjects or personal data.

REFERENCES

Daniel Adelman and Adam J Mersereau. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, 56(3):712–727, 2008.

James H Anderson and Anand Srinivasan. Mixed pfair/erfair scheduling of asynchronous periodic tasks. *Journal of Computer and System Sciences*, 68(1):157–204, 2004.

Shoshana Anily and Julien Bramel. Periodic scheduling with service constraints. *Operations Research*, 48(4):635–645, 2000.

Heesang Ann, Hyunjun Choi, Taehyun Hwang, Younghoon Shin, Haeju Cheong, and Min-hwan Oh. Generalized linear bandits with memory. *arXiv preprint arXiv:2608.15848*, 2026.

Alexia Atsidakou, Orestis Papadigenopoulos, Soumya Basu, Constantine Caramanis, and Sanjay Shakkottai. Combinatorial blocking bandits with stochastic delays. In *International Conference on Machine Learning*, pp. 404–413. PMLR, 2021.

Amotz Bar-Noy, Randeep Bhatia, Joseph Naor, and Baruch Schieber. Minimizing service and operation costs of periodic scheduling. *Mathematics of Operations Research*, 27(3):518–544, 2002.

Sanjoy K Baruah, Neil K Cohen, C Greg Plaxton, and Donald A Varvel. Proportionate progress: A notion of fairness in resource allocation. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pp. 345–354, 1993.

Soumya Basu, Rajat Sen, Sujay Sanghavi, and Sanjay Shakkottai. Blocking bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

Dimitris Bertsimas and José Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48(1):80–90, 2000.

David B Brown and Jingwei Zhang. Fluid policies, reoptimization, and performance guarantees in dynamic resource allocation. *Operations Research*, 73(2):1029–1045, 2025.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback–Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013. doi: 10.1214/13-AOS1119.

Leonardo Cella and Nicolo Cesa-Bianchi. Stochastic bandits with delay-dependent payoffs. In *International Conference on Artificial Intelligence and Statistics*, pp. 1168–1177. PMLR, 2020.

Nicolo Cesa Bianchi, Junya Honda, Yuko Kuroki, Atsushi Miyauchi, Lukas Zierahn, et al. Learning periodic strategies in blocking bandits is as hard as bandits with switching costs. *PROCEEDINGS OF MACHINE LEARNING RESEARCH*, 336:1001–1021, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026.

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.

Daniel Freund and Rowan Hess. A general algorithm for online resource allocation with recharging rewards. *Available at SSRN 6990698*, 2026.

Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006. doi: 10.1145/1147954.1147956.

Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pp. 359–376. JMLR Workshop and Conference Proceedings, 2011.

Wassily Hoeffding. On the distribution of the number of successes in independent trials. *The Annals of Mathematical Statistics*, 27(3):713–721, 1956. doi: 10.1214/aoms/1177728178.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Anand Kalvit and Divya Singhvi. Online learning for repeated nudging. *NYU Stern School of Business Research Paper*, 2025.

Robert Kleinberg and Nicole Immorlica. Recharging bandits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 309–319. IEEE, 2018.

Donald E. Knuth. Two-way rounding. *SIAM Journal on Discrete Mathematics*, 8(2):281–290, 1995. doi: 10.1137/S0895480194264757.

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*, 355, 2019.

Dongfang Li, Zixuan Liu, Junmai Wang, Jiahe Huang, Fuhao Li, Bonian Jia, Baotian Hu, and Min Zhang. Lycheememory v2: Efficient long-term memory for llm agents via semantic segment-level consolidation. *arXiv preprint arXiv:2608.12990*, 2026.

Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. A finite-time analysis of multi-armed bandits problems with kullback-leibler divergences. In *Proceedings of the 24th annual conference on learning theory*, pp. 497–514. JMLR Workshop and Conference Proceedings, 2011.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Orestis Papadigenopoulos, Constantine Caramanis, and Sanjay Shakkottai. Non-stationary bandits under recharging payoffs: Improved planning with sublinear regret. *Advances in Neural Information Processing Systems*, 35:20325–20337, 2022.

Ciara Pike-Burke and Steffen Grunewalder. Recovering bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

David Simchi-Levi, Zeyu Zheng, and Feng Zhu. Dynamic planning and learning under recovering rewards. In *International Conference on Machine Learning*, pp. 9702–9711. PMLR, 2021.

David Simchi-Levi, Zeyu Zheng, and Feng Zhu. Offline planning and online learning under recovering rewards. *Management Science*, 71(1):298–317, 2025.

Mika Sumida. Dynamic resource allocation with recovering rewards under non-stationary arrivals. *USC Marshall School of Business Research Paper Sponsored by iORB*, 2024.

Yuto Tanimoto and Kenji Fukumizu. State-separated sarsa: A practical sequential decision-making algorithm with recovering rewards. *Algorithms*, 19(5):419, 2026.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345, 2024.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

Kevin P Yancey and Burr Settles. A sleeping, recovering bandit algorithm for optimizing recurring notifications. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3008–3016, 2020.

Xiangcheng Zhang, Yige Hong, and Weina Wang. Projection-based lyapunov method for fully heterogeneous weakly-coupled mdps. *Advances in Neural Information Processing Systems*, 38: 91054–91076, 2026.

A Proof of Theorem 1

Normalize $L = 1$ by dividing each reward curve by L. We construct a feasible calendar from a renewal plan and bound the loss introduced by its implementation.

We begin with the renewal LP. Fix initial ages a, an integer $h \geq 1$, and any feasible h-date calendar C. Let N_{ip} count activations of arm i, other than its first, whose preceding service gap is p. These completed gaps are disjoint for each arm and lie within the calendar. Their durations and the total pull count therefore satisfy:

$$\sum_p pN_{ip} \leq h \quad (i \in [n]), \qquad \sum_{i,p} N_{ip} \leq Kh.$$

Setting $x_{ip} = pN_{ip}/h$ therefore gives a feasible solution of (3). Each omitted first activation earns at most one, so

$$\begin{aligned} \frac{J_R(C;a)}{h} &\leq \frac{n}{h} + \sum_{i,p} R_i(p)\frac{N_{ip}}{h} \\ &= \frac{n}{h} + \sum_{i,p} \frac{R_i(p)}{p} x_{ip} \leq \frac{n}{h} + \mathrm{UB}(R). \end{aligned}$$

Taking long-run averages yields $\mathrm{OPT} \leq \mathrm{UB}$.

Next, we make the LP finite. Let UB_Q be (3) restricted to $p \leq Q$. The omitted columns of any feasible solution satisfy

$$\sum_{i,p>Q} \frac{R_i(p)}{p} x_{ip} \leq \frac{1}{Q} \sum_{i,p>Q} x_{ip} \leq \frac{n}{Q},$$

so

$$0 \leq \mathrm{UB} - \mathrm{UB}_Q \leq n/Q. \tag{A.1}$$

The finite LP has the basic-solution structure in Lemma 1.

Proof of Lemma 1. Choose an optimal basic solution of this finite LP. Add an idle variable $x_{i\infty}$ to each arm constraint and a capacity slack z:

$$\sum_{p \in [Q] \cup \{\infty\}} x_{ip} = 1 \quad (i \in [n]), \qquad \sum_{i,p \leq Q} \frac{x_{ip}}{p} + z = K.$$

There are $n+1$ equalities. If s_i counts the positive choices of arm i, including idling, a basic solution therefore satisfies

$$s_i \geq 1, \qquad \sum_i s_i \leq n+1 \quad \Longrightarrow \quad \sum_i (s_i - 1) \leq 1.$$

All but at most one arm use a single period or idle. The exceptional arm mixes at most two choices. The same argument applies when $[Q]$ is replaced by any finite period set. □

For the exceptional arm, sample P_i once per candidate block, with $\Pr(P_i = p) = x_{ip}$. Conditional on $P_i < \infty$, sample a uniform phase modulo P_i. Its request indicator $I_i(t)$ obeys

$$\mathbb{E}I_i(t) = \sum_{p \leq Q} \frac{x_{ip}}{p}, \qquad \mathbb{E}[I_i(t)R_i(P_i)] = \sum_{p \leq Q} \frac{x_{ip}R_i(p)}{p},$$

where the idle draw contributes zero. These equalities preserve its LP frequency and reward rate in expectation. Keep this arm in the head, regardless of its sampled period; the other streams are built from arms assigned a single period.

A.1 LOSSLESS AGGREGATION AND SHORT-PERIOD COORDINATION

We exploit divisibility to combine periodic plans while preserving each arm's waiting interval. An aggregated stream generates one request every p rounds, following a fixed repeating sequence of arm labels. If arm i appears every m_i requests, its nominal wait and reward are

$$p_i^0 = m_i p, \qquad v_i = R_i(p_i^0), \qquad \sum_i \frac{1}{m_i} = 1.$$

Each request is assigned to exactly one arm, giving the last equality. Initially each stream contains one arm, with $m_i = 1$. Write $\mathrm{spf}(p)$ for the smallest prime factor of p. The recursive compression below is denoted Aggregate.

Lemma A.1 (Lossless stream compression)**.** *Cycling through k streams of period p, where $k \mid p$, gives a single stream of period p/k with the same nominal gap and reward rate for every arm. Processing periods $p \geq 2$ in decreasing order, and merging in groups of $\mathrm{spf}(p)$, leaves residual counts*

$$0 \leq r_p < \mathrm{spf}(p). \tag{A.2}$$

Each period-one stream occupies one complete daily slot.

Proof. For an arm in any merged stream, the transformation is

$$(p, m_i) \longmapsto (p/k, km_i), \qquad \underbrace{(km_i)(p/k)}_{\text{new gap}} = m_i p, \qquad \frac{v_i}{(km_i)(p/k)} = \frac{v_i}{m_i p}.$$

Thus both frequency and nominal reward rate are unchanged. At period p, merge until fewer than $\mathrm{spf}(p)$ streams remain. Every new stream has a smaller period, so later operations cannot add streams at a period already processed. This proves (A.2). A period-one stream assigns one arm to each date and therefore occupies one complete slot. □

The remaining loss analysis concerns the residual streams in (A.2); complete daily slots retain their nominal rewards.

For short periods, phase coordination uses the remaining divisibility structure without reducing request rates. Partition residual periods $2, \ldots, 16$ as

$$\{2, 4, 8, 16\}, \quad \{3, 6, 9, 12\}, \quad \{5, 10, 15\}, \quad \{7, 14\}, \quad \{11\}, \quad \{13\}.$$

Each package has an independent uniform overall time phase; each stream also starts at an independent uniform position in its repeating arm sequence. For a divisibility chain, assign streams successively to free residue classes of their periods. The first, fourth, fifth, and sixth packages then have at most one request per date.

In the mod-three package, the $r_3 \leq 2$ period-three streams occupy distinct residue classes. The period-six and period-twelve streams fit, without overlap, inside one remaining class; the period-nine streams occupy distinct subresidues inside another remaining class when available. If $r_3 \leq 1$, these subpackages do not overlap and the total is Bernoulli. If $r_3 = 2$, they share the sole remaining class. Write

$$u = r_6/2 + r_{12}/4 \in \{0, 1/4, 1/2, 3/4\}, \qquad v = r_9/3 \in \{0, 1/3, 2/3\}.$$

Uniform phase modulo 36 makes the mod-four and mod-three subresidues independent. The package count X_3 has probabilities

$$\Pr(X_3 = 0) = (1-u)(1-v)/3, \quad \Pr(X_3 = 2) = uv/3,$$

with the remaining probability at one. There are twelve such laws and the Bernoulli case. The mod-five package is identical in form: if $r_5 \leq 3$ it is Bernoulli; otherwise use $u = r_{10}/2 \in \{0, 1/2\}$, $v = r_{15}/3 \in \{0, 1/3, 2/3\}$ and replace the denominator three by five. This gives six fixed laws and one Bernoulli case.

The total head request count N is a sum of five independent Bernoulli variables (including the exceptional LP arm), plus independent X_3 and X_5 of the stated types. A missing package is a zero Bernoulli. Write $\mu = \mathbb{E}N$. We will bound head overflow using these package laws and the capacity left by the tail.

Residual periods $p \geq 17$ form the tail. The next construction applies to any tail group; Appendix A.3 chooses its periods and buffer.

A.2 BUFFERED STREAMS AND GAP-SAFE TRANSPORT

We assign requests to service slots in order and bound the number removed by the waiting-time test. The source stream determines the request order; independently constructed quotas determine the available slots.

Fix a nonempty group of streams, with r_p streams at each occupied period p. The streams may already have undergone upward period replacements: each arm has a target gap $d_i = m_i p \geq p_i^0$. The original gap p_i^0 determines its nominal reward v_i, and d_i is the threshold for the retention test. Appendix A.3 specifies the replacements and groups. Add a common buffer $\delta > 0$ between successive requests of each stream, and write

$$\tau_p = p + \delta, \qquad \lambda = \sum_p r_p/p, \qquad \rho = \sum_p r_p/\tau_p, \qquad F = \delta\rho.$$

Here λ is the group's unbuffered request rate and ρ its buffered rate. We call the collection of its r_p equal-period streams pool p. Cycling through these streams every τ_p/r_p source-time units makes the pool regularly spaced. Arm i appears every $r_p m_i$ pool requests, so its buffered source gap is $m_i\tau_p$. Hence

$$\rho m_i \tau_p = \rho d_i + m_i F.$$

Thus $m_i F$ is the additional expected separation in request ranks.

To make the arm-label distribution the same at every rank, choose a pool P^* in proportion to its rate, place one of its requests at time zero, and choose its position in the repeating arm sequence uniformly:

$$\Pr(P^* = p) = \frac{r_p/\tau_p}{\rho}.$$

Independently sample each other pool's time phase on a common rational tick and its sequence position uniformly. Order ties by a fixed pool order; requests preceding the chosen request, including earlier ties at zero, have negative indices.

Let d_g be the number of occupied periods in this group. The following lemma describes the label and preceding same-arm gap at any request rank.

Lemma A.2 (Pointed-stream law)**.** *Under the preceding sampler, the merged label sequence $(\ell_k)_{k\in\mathbb{Z}}$ is stationary under integer rank shifts. At every fixed rank, arm i in pool p has probability*

$$\pi_i = \frac{1/(m_i\tau_p)}{\rho}. \tag{A.3}$$

Conditional on that arm, the rank distance M_i to its preceding request has mean $\rho d_i + m_i F$. Moreover, $Y_i := \mathbb{E}M_i - M_i$ is a sum of at most $d_g - 1$ independent centered variables, each with range at most one.

Proof. Consider one common period of all pool times and arm sequences. A uniform request belongs to pool p with probability $(r_p/\tau_p)/\rho$; conditional on that request, the other phases remain independent and uniform. Moving to the next request permutes the choices of distinguished request. Hence

$$(\ell_{k+r})_{k\in\mathbb{Z}} \overset{d}{=} (\ell_k)_{k\in\mathbb{Z}} \qquad (r \in \mathbb{Z}).$$

The direct phase sampler realizes this law through pool phases and sequence positions. Uniform sequence positions preserve the arm frequencies, giving (A.3).

Conditional on arm i in pool p, the own-pool count is $r_p m_i$. Each other pool contributes an independent floor-or-ceiling count over length $m_i\tau_p$. Therefore

$$\mathbb{E}M_i = r_p m_i + \sum_{q\neq p} \frac{r_q}{\tau_q} m_i \tau_p = \rho m_i \tau_p = \rho d_i + m_i F,$$

$$Y_i := \mathbb{E}M_i - M_i = \sum_{q\neq p} \{\mathbb{E}M_{iq} - M_{iq}\}.$$

Here M_{iq} is the count from pool q, and each centered count has range at most one. There are at most $d_g - 1$ such pools. □

Rank stationarity lets us apply the same source law at any destination slot. We now construct these slots independently of the source phases. The quotas preserve per-round capacity and bound the number of slots within any waiting-time window. We use a time-chain version of marginal-preserving dependent rounding (Gandhi et al., 2006), related to fair scheduling and two-way rounding (Baruah et al., 1993; Anderson & Srinivasan, 2004; Knuth, 1995).

Lemma A.3 (Dependent quotas and integer-window control)**.** *Let rational rates $a_g \geq 0$, $g \in [G]$, sum to an integer M, and fix an integer $N_0 \geq 1$. There is a randomized construction of quotas on dates $1, \ldots, N_0$ satisfying*

$$\sum_g C_g(t) = M, \qquad C_g(t) \in \{\lfloor a_g \rfloor, \lceil a_g \rceil\}, \qquad \mathbb{E}[C_g(t) \mid R] = a_g. \tag{A.4}$$

Here R is uniform modulo a common denominator of the rates, and $C_g(t) = Q_g(R+t) - Q_g(R+t-1)$. For $0 \leq t \leq N_0$,

$$Q_g(R+t) \in \{\lfloor a_g(R+t) \rfloor, \lceil a_g(R+t) \rceil\}, \quad \mathbb{E}[Q_g(R+t) \mid R] = a_g(R+t).$$

Consequently each prefix error $e_g(t) = Q_g(R+t) - a_g(R+t)$ has conditional mean zero and, given R, lies in an interval of length at most one. For integers $0 \leq t-d \leq t \leq N_0$ and z, put $D = Q_g(R+t) - Q_g(R+t-d)$. Then

$$\mathbb{E}(D-z)_+ \leq \mathbb{E}(a_g d - z + V)_+, \qquad V \sim \mathrm{Unif}[-1,1]. \tag{A.5}$$

The construction takes $O(GN_0)$ flow-rounding updates.

Proof. We first construct the cumulative quotas, then bound a window count. Remove integer bases: $\beta_g = a_g - \lfloor a_g \rfloor$ and $M_{\mathrm{frac}} = \sum_g \beta_g \in \mathbb{Z}$. A date node supplies M_{frac} units through group-date arcs with bounds $[0,1]$. Time-chain arcs carry cumulative flow within

$$[\lfloor \beta_g(R+t) \rfloor, \lceil \beta_g(R+t) \rceil].$$

An initial node supplies $M_{\mathrm{frac}} R$ through group arcs with bounds $[\lfloor \beta_g R \rfloor, \lceil \beta_g R \rceil]$; terminal arcs enter a common sink. Initial fractional flows $\beta_g R$ and per-date flows β_g form a feasible flow.

Let f be the current flow and v an oriented circulation in its noninteger support. If ϵ_+, ϵ_- are the distances to the nearest integer boundary in the two directions, update

$$f' = \begin{cases} f + \epsilon_+ v, & \text{with probability } \epsilon_-/(\epsilon_+ + \epsilon_-), \\ f - \epsilon_- v, & \text{with probability } \epsilon_+/(\epsilon_+ + \epsilon_-). \end{cases} \qquad \mathbb{E}[f' \mid f] = f.$$

Flow and bounds are preserved, and at least one additional arc becomes integral. A nonempty fractional support cannot be a forest: a leaf node would have one noninteger arc but an integer node balance. Thus at most $O(GN_0)$ updates produce an integral flow with the original marginals. Restoring the integer bases proves (A.4), the cumulative rounding properties, and the stated conditional properties of $e_g(t)$.

For the window bound, fix a group and write $a = a_g$, $Q = Q_g$. Write $ad = k + \alpha$, $k \in \mathbb{Z}$, $0 \leq \alpha < 1$, and $u = \{a(R+t)\}$, $v = (u - \alpha) \bmod 1$. Conditional on R, the two rounding indicators have Bernoulli marginals u, v, possibly dependent. Hence

$$\begin{aligned} u \geq \alpha: &\quad \Pr(D = k-1 \mid R) \leq \min(u - \alpha, 1 - u), \\ u < \alpha: &\quad \Pr(D = k+2 \mid R) \leq \min(u, \alpha - u). \end{aligned}$$

The origin makes u uniform on the denominator grid, with α on that same grid. Each triangular function has grid-aligned endpoints; its grid average is at most its integral. Hence

$$\Pr(D = k-1) \leq (1-\alpha)^2/4, \qquad \Pr(D = k+2) \leq \alpha^2/4.$$

Since $D \in \{k-1, k, k+1, k+2\}$ and $\mathbb{E}D = k + \alpha$,

$$\mathbb{E}(D-k)_+ = \alpha + \Pr(D = k-1) \leq \frac{(1+\alpha)^2}{4} = \mathbb{E}(\alpha + V)_+,$$

$$\mathbb{E}(D-k-1)_+ = \Pr(D = k+2) \leq \frac{\alpha^2}{4} = \mathbb{E}(\alpha - 1 + V)_+.$$

Lower integer thresholds are linear and higher ones give zero, proving (A.5). □

Apply this construction jointly to all retained groups, with mean $a_g = \rho_g$ for each tail group and the remaining mean quota assigned to the head. Appendix A.3 verifies this allocation. Joint rounding preserves the per-round sum of all group quotas.

Include $d_{\max} := \max_i d_i$ dates of virtual quota history so that the waiting-time windows are defined at the start of the calendar. Physical execution still starts at the inherited ages. The uniform variable in (A.5) is used only to bound window-count fluctuations.

Reindex this group's cumulative quotas so that $Q(0) = 0$, and assign request k to slot k. Its date and retention rule are

$$t_k = \min\{t : Q(t) \geq k\}, \qquad k^- = \max\{r < k : \ell_r = \ell_k\}, \qquad \text{retain } k \iff t_k - t_{k^-} \geq d_{\ell_k}. \tag{A.6}$$

Here k^- is the preceding *raw* request, whether retained or not. The quota enforces capacity; the retention test enforces the required same-arm gap. Only the preceding $d_{\max}$ dates matter; an earlier predecessor automatically passes this test.

The next lemma converts these rank and quota laws into a feasible calendar and a bound on deleted requests.

Lemma A.4 (Gap-safe transport and slotwise loss)**.** *Apply (A.6) to a stream from Lemma A.2, independently of its quotas from Lemma A.3. Retained requests use only the assigned slots, activate each arm at most once per date, and earn at least their nominal rewards after each arm's first real activation. If $N_{\mathrm{del}}(t)$ counts deletions on date t, then*

$$\mathbb{E}N_{\mathrm{del}}(t) \leq \sum_i \pi_i \, \mathbb{E}(D_{t,i} - M_i)_+, \qquad D_{t,i} = Q(t) - Q(t - d_i). \tag{A.7}$$

Here M_i has the conditional rank-gap law given arm i and is independent of the quotas.

Proof. Deletion never stops the raw stream. For two successive retained activations of arm i, after its first real activation,

$$d_i^{\mathrm{actual}} \geq t_k - t_{k^-} \geq d_i \geq p_i^0 \quad \Longrightarrow \quad R_i(d_i^{\mathrm{actual}}) \geq v_i.$$

A second same-date request has raw gap zero and is removed. Quotas bound the number of used slots. Charging the first real activation once per arm handles arbitrary inherited ages; the virtual history does not reset them.

We now count deletions. For the rth slot backward from the end of date t, its predecessor is too close exactly when

$$Q(t) - r - M_i > Q(t - d_i) \quad \Longleftrightarrow \quad M_i \leq D_{t,i} - r - 1.$$

Condition on the entire quota sequence $\mathcal{Q}$. Each selected rank still has arm-label distribution π, by independence and rank stationarity. Hence

$$\begin{aligned}
\mathbb{E}[N_{\mathrm{del}}(t) \mid \mathcal{Q}] &= \sum_{r=0}^{C(t)-1} \sum_i \pi_i \Pr(M_i \leq D_{t,i} - r - 1 \mid i, \mathcal{Q}) \\
&\leq \sum_i \pi_i \sum_{r \geq 0} \Pr(M_i \leq D_{t,i} - r - 1 \mid i, \mathcal{Q}) \\
&= \sum_i \pi_i \, \mathbb{E}[(D_{t,i} - M_i)_+ \mid i, \mathcal{Q}].
\end{aligned}$$

Averaging proves (A.7). Since $\sum_i \pi_i = 1$, the bound gives a single loss charge per group and round. □

A.3 Multiscale buffering and coupled loss accounting

We now choose period groups and buffers for the residual streams. A larger buffer reduces deletions but lowers the request rate; longer periods make that rate reduction cheaper. To reduce the number of distinct periods, replace each residual period $p \geq 17$ by

$$q(p) = \min\{q \geq p : q \in \mathcal{A}\}, \qquad \mathcal{A} = \{q \geq 1 : \gcd(q, 30) = 1\}.$$

Preserve the order of arms within each aggregated stream. Monotonicity gives

$$m_i q(p) \geq m_i p = p_i^0 \quad \Longrightarrow \quad R_i(m_i q(p)) \geq v_i.$$

Merge every q streams now sharing period q into a complete daily slot; the remaining count is $r_q < q$. Unlike the compression in Appendix A.1, period replacement lowers service frequency. Using the pre-replacement counts in this debit,

$$\delta_0 = \sum_{p \geq 17} r_p (1/p - 1/q(p)).$$

After this step, r_p will denote the residual count at the new period. An excluded period has smallest prime factor at most five, so (A.2) bounds its multiplicity by four. Consecutive admissible periods are at most six apart; a moved stream therefore shifts by at most five dates.

Every transformed arm gap is smaller than $2Q$, so the assignment requires fewer than $2Q$ virtual history dates.

For any group whose periods are at least P, the first cost of buffering has the simple bound

$$\lambda - \rho = \delta \sum_p \frac{r_p}{p(p+\delta)} \leq \frac{\delta}{P} \sum_p \frac{r_p}{p+\delta} = \frac{F}{P}.$$

The same rank buffer therefore costs less at longer periods. For the second cost, let $X_i = D_{t,i} - \rho d_i$ be the quota-window error. The pointed-stream law gives

$$D_{t,i} - M_i = X_i + Y_i - m_i F.$$

Premature services occur only when the combined fluctuation exceeds the buffer. We bound this excess with a finite binomial calculation for near groups and an exponential estimate for far groups. Define

$$\mathcal{M}_m(s) = \sup_{0 \leq p \leq 1} \mathbb{E}[\mathrm{Bin}(m,p) - mp - s]_+, \qquad \Theta_m(F) = \frac{1}{2} \int_{-1}^{1} \mathcal{M}_m(F - v)\, dv.$$

Lemma A.5 (Buffered deletion bound)**.** *For a retained group with d_g occupied periods and buffer F, the transport in Lemma A.4 satisfies*

$$\mathbb{E}N_{\mathrm{del}}(t) \leq \Theta_{d_g - 1}(F).$$

If $F \geq \sqrt{(d_g+3)b}$ for some $b > 0$, it also satisfies

$$\mathbb{E}N_{\mathrm{del}}(t) \leq \frac{\sqrt{d_g + 3}}{4e\sqrt{b}}\, e^{-2b}.$$

Proof. For the first bound, we control the rank fluctuation by a binomial envelope and then average over the quota-window error. Endpoint replacement for convex functions, followed by equalizing Bernoulli parameters of fixed sum, bounds a sum of m independent centered range-one variables by this envelope (Hoeffding, 1956). Indeed, conditional on the other summands, the coefficient of $p_1 p_2$ in the expectation of a convex function φ is

$$\varphi(x+2) - 2\varphi(x+1) + \varphi(x) \geq 0.$$

At fixed $p_1 + p_2$, averaging the two parameters increases their product. Iterating gives the binomial envelope.

Apply (A.5) conditional on the integer M_i. The source and quota are independent, so

$$\mathbb{E}(D_{t,i} - M_i)_+ \leq \mathbb{E}(Y_i - m_i F + V)_+ \leq \Theta_{d_g - 1}(F).$$

Together with (A.7) and $\sum_i \pi_i = 1$, this bounds a group's daily deletion count by $\Theta_{d_g-1}(F)$. Increasing the buffer therefore reduces the deletion charge through a single scalar argument.

For the second bound, write the quota-window error as $X = D_{t,i} - \rho d_i = e(t) - e(t - d_i)$, where e is this group's prefix error from Lemma A.3. Conditional on R, each prefix error is centered and has range at most one. Cauchy–Schwarz and the bounded-variable exponential bound give

$$\mathbb{E}[e^{\eta X} \mid R] \leq \{\mathbb{E}[e^{2\eta e(t)} \mid R]\mathbb{E}[e^{-2\eta e(t-d_i)} \mid R]\}^{1/2} \leq e^{\eta^2/2},$$
$$\mathbb{E}e^{\eta Y_i} \leq e^{(d_g - 1)\eta^2/8}.$$

Using source–quota independence and $x_+ \le e^{\eta x}/(e\eta)$,

$$\mathbb{E}(Y_i + X - F)_+ \le \frac{1}{e\eta} \exp\left\{ -\eta F + \frac{d_g + 3}{8}\eta^2 \right\}.$$

For $F \ge \sqrt{(d_g+3)b}$, choose $\eta = 4\sqrt{b/(d_g+3)}$ to obtain

$$\mathbb{E}(Y_i + X - F)_+ \le \frac{\sqrt{d_g+3}}{4e\sqrt{b}}\, e^{-2b}. \tag{A.8}$$

Since $m_i \ge 1$, the positive part in (A.7) is bounded by the left-hand side of (A.8). Summing with weights π_i proves the second assertion. □

We can now choose the groups and buffers. Use three near groups

$$[17, 53), \qquad [53, 149), \qquad [149, 449),$$

followed by the far groups $[P_j, 3P_j)$ with $P_j = 449 \cdot 3^j$, $j \ge 0$. The finite near groups receive precise rate budgets; the far groups use a common rule whose two costs are summable across scales.

Some groups contain too few streams to attain the chosen buffer. For a group starting at P, let $W = \sum_p r_p$, let f be its buffer target, and set the approximation tolerance $e_P = P^{-4}$. Use a margin factor $\Gamma > 1$ and delete the group if $W \le \Gamma(f + e_P)$. Otherwise choose δ satisfying

$$f \le \delta \sum_p \frac{r_p}{p+\delta} \le f + e_P. \tag{A.9}$$

Indeed,

$$F(\delta) = \delta \sum_p \frac{r_p}{p+\delta} \quad \text{increases from 0 to } W.$$

Thus the target is attainable for retained groups; deleted groups are charged their full request rate. The fixed parameters and rational implementation are given in Appendix A.5.

For the near groups, we choose a buffer from a prescribed rate-loss budget. The conversion depends on a weighted inverse period. Let $\mathcal{P}_j = \mathcal{A} \cap [P_j, P_{j+1})$, where $(P_0, P_1, P_2, P_3) = (17, 53, 149, 449)$, and define

$$h_{j,d} = \max_{\substack{S \subseteq \mathcal{P}_j,\ |S| = d \\ 1 \le r_p \le p-1}} \frac{\sum_{p \in S} r_p/p^2}{\sum_{p \in S} r_p/p}.$$

For rate budgets $c_j > 0$ and grid size ξ, take

$$f_{j,d} = \xi \left\lfloor \frac{c_j}{\xi h_{j,d}} \right\rfloor. \tag{A.10}$$

To justify this choice, the rate debit of a retained group satisfies

$$\lambda - \rho \le F \frac{\sum_p r_p/p^2}{\sum_p r_p/p}. \tag{A.11}$$

To see this, let $w_p(\delta) = (r_p/(p+\delta))/\rho$. Then

$$\frac{\lambda - \rho}{F} = \sum_p \frac{w_p(\delta)}{p}, \qquad \frac{d}{d\delta} \sum_p \frac{w_p(\delta)}{p} = -\operatorname{Cov}_{w(\delta)}\left(\frac{1}{p}, \frac{1}{p+\delta} \right) \le 0.$$

Its value at zero proves (A.11). For a deleted group, Cauchy–Schwarz gives

$$\lambda^2 = \left(\sum_p \frac{r_p}{p} \right)^2 \le W \sum_p \frac{r_p}{p^2} \quad \Longrightarrow \quad \frac{\lambda}{W} \le \frac{\sum_p r_p/p^2}{\sum_p r_p/p}.$$

Thus both branches have rate loss at most

$$\Gamma(f_{j,d} + e_{P_j}) h_{j,d} \le \Gamma(c_j + P_j^{-5}).$$

The budget c_j therefore controls the rate loss uniformly over the near group's support, including the case in which the group is deleted.

For a far group starting at $P = 449 \cdot 3^j$, use $b_j = b_* + \kappa j$ and a buffer target $f \geq \sqrt{(d_g+3)b_j}$ satisfying $f + e_P \leq \Gamma\{\sqrt{(d_g+3)b_j} + e_P\}$. The rate loss in either branch is therefore bounded by

$$\frac{\Gamma^2}{P}\{\sqrt{(d_g+3)b_j} + P^{-4}\}. \tag{A.12}$$

For a retained group, the deletion bound is (A.8). Since d_g counts occupied periods in $[P, 3P)$, $d_g = O(P)$. The two bounds consequently have respective orders

$$O\left(\frac{\sqrt{j+1}}{3^{j/2}}\right), \qquad O\left(\frac{(\sqrt{3}e^{-2\kappa})^j}{\sqrt{j+1}}\right).$$

Both are summable when $\kappa > (\log 3)/4$. Progressively longer periods can thus absorb larger buffers cheaply enough to make the total charge finite across all period scales.

Let R_+ and E_+ be rational upper bounds on the total rate loss (including δ_0) and the total daily deletion charge, respectively. Appendix A.5 constructs these bounds from the preceding estimates.

Reducing the tail rate releases capacity for short-period requests. We therefore bound both costs using the same capacity slack.

After complete daily slots have been removed, write M for the remaining capacity, μ for the head mean, and λ_T for the tail nominal rate after period replacement. Let Δ be the tail rate reduction, including whole-group deletions. Assign quota means

$$a_g = \rho_g \quad \text{for tail group } g, \qquad q_H = M - \sum_g \rho_g = M - \lambda_T + \Delta.$$

Original LP feasibility now gives the shared slack:

$$\mu + \lambda_T + \delta_0 \leq M \quad \Longrightarrow \quad q_H - \mu \geq \delta_0 + \Delta =: s, \qquad 0 \leq s \leq R_+.$$

The head count N is integer, and its quota C_H has adjacent-integer support with mean q_H. Independence of source phases and quotas yields

$$\mathbb{E}(N - C_H)_+ = \mathbb{E}(N - q_H)_+ \leq F_N(s), \qquad F_N(s) = \mathbb{E}(N - \mu - s)_+.$$

The rate reduction and head overflow share the same capacity slack:

$$s + F_N(s) = \mathbb{E}\max\{s, N - \mu\}. \tag{A.13}$$

The combined expression is nondecreasing in s, so one upper bound on s controls both costs.

The finite calculation in Appendix A.5 supplies a uniform bound $\mathcal{F}(s) \geq F_N(s)$ for every head law from Appendix A.1. For a rational grid size $\nu > 0$, set $\bar{s} = \nu\lceil R_+/\nu \rceil$. Equation (A.13) and the tail estimates give the uniform loss bound

$$c_{\text{loss}} := \bar{s} + \mathcal{F}(\bar{s}) + E_+. \tag{A.14}$$

Here c_{loss} bounds the expected nominal-reward loss per date, uniformly over the LP allocation.

A.4 Calendar certificate and long-run guarantee

Algorithm A.1 collects the construction at the original reward scale L. Each trial uses $2Q$ virtual dates and $Q_g(0) = 0$; a predecessor outside this history has date $-\infty$. Set $v_i = 0$ for idle arms.

Algorithm A.1 Multiscale Buffered Interleaving (MBI)

Input: n, K, L, R, a, h; optionally, an optimal basic solution x^* of the LP on $p \le 100n$.

Obtain x^* and $\mathrm{UB}_Q(R)$ from LP (3), using the supplied x^* if present.
Aggregate in decreasing $p \ge 2$ with $m = \mathrm{spf}(p)$. ▷ Step 1
$p \leftarrow q(p)$ for residual $p \ge 17$; extract complete slots, leaving $r_p < p$. ▷ Step 2
Form the groups and targets f of Appendix A.3.
Retain groups with $W > \Gamma(f + e_P)$; choose each δ by (A.9).
$a_g \leftarrow \rho_g$, $q_H \leftarrow M - \sum_g \rho_g$.
repeat ▷ Step 3
  Exceptional arm, if any: draw $P_i \sim x_i^*$; set $v_i \leftarrow R_i(P_i)$ (idle: $v_i \leftarrow 0$).
  Sample head phases as above and independent uniform complete-slot sequence phases.
  Independently sample each tail sequence $(\ell_k)_{k\in\mathbb{Z}}$ by Lemma A.2.
  Sample joint quotas $C_H(t), C_g(t), Q_g(t)$ by Lemma A.3, independently of requests.
  Initialize each C_t with complete-slot activations and up to $C_H(t)$ head requests in arm order.
  for each retained group g **do**
    $t_k \leftarrow \min\{t : Q_g(t) \ge k\}$, $k^- \leftarrow \max\{r < k : \ell_r = \ell_k\}$.
    $C_{t_k} \leftarrow C_{t_k} \cup \{\ell_k\}$ for $1 \le t_k \le h$ with $t_k - t_{k^-} \ge d_{\ell_k}$.
  end for
  $Z \leftarrow \sum_{i=1}^{n} v_i\big(|\{t : i \in C_t\}| - 1\big)_+$.
until $Z \ge [h\{\mathrm{UB}_Q(R) - 99L/100\} - nL]_+$
return C, Z

Completion of the proof of Theorem 1. The head quotas and Lemma A.4 ensure feasibility. Each deletion costs at most one, and omitting each arm's first real activation costs at most n in total. Equations (A.7) and (A.14) therefore give, for any inherited ages,

$$0 \le Z \le J_R(C; a) \le hK, \qquad \mathbb{E}Z \ge h\{\mathrm{UB}_Q(R) - c_{\mathrm{loss}}\} - n. \tag{A.15}$$

Take $Q = 100n$. In the normalization $L = 1$, write $S_{\mathrm{poly}}(R) = [\mathrm{UB}_Q(R) - 99/100]_+$ and $a_h(R) = [hS_{\mathrm{poly}}(R) - n]_+$. The LP-based score S_{poly} is coordinatewise nondecreasing, and (A.1) gives $S_{\mathrm{poly}}(R) \ge [\mathrm{UB}(R) - 1]_+$. Appendix A.5 establishes $c_{\mathrm{loss}} < 97/100$. If $a_h(R) = 0$, every candidate passes. Otherwise (A.15) and $Z \le hK$ imply

$$\frac{h}{50} \le h\left(\frac{99}{100} - c_{\mathrm{loss}}\right) \le \mathbb{E}Z - a_h(R) \le hK \Pr\{Z \ge a_h(R)\}.$$

Returning the first independent passing candidate therefore gives

$$\Pr(\text{pass}) \ge \frac{1}{50K}, \qquad \mathbb{E}[\text{trials}] \le 50K.$$

Restoring the original reward scale, the score and acceptance threshold are

$$S_{\mathrm{poly}}(R) = [\mathrm{UB}_Q(R) - 99L/100]_+, \qquad a_h(R) = [hS_{\mathrm{poly}}(R) - nL]_+, \tag{A.16}$$

and every returned calendar satisfies

$$J_R(C; a) \ge a_h(R) \ge \big[h\{\mathrm{UB}(R) - L\} - nL\big]_+. \tag{A.17}$$

For the long-run guarantee, concatenate blocks of lengths $h_k = k$. After m completed blocks, even allowing an unfinished next block, the average initialization and boundary charge, in units of L, is at most

$$\frac{mn + K(m+1)}{\sum_{k=1}^{m} k} = \frac{2\{mn + K(m+1)\}}{m(m+1)} \longrightarrow 0.$$

The block concatenation gives $\mathrm{UB}(R) - L \le \mathrm{OPT}(R) \le \mathrm{UB}(R)$, completing the reward guarantees. □

A.5 Parameter certificate and computation

We evaluate c_{loss} with fixed parameters, leaving enough slack for LP truncation and candidate acceptance, and specify a compact implementation of the construction.

Period-rounding loss. For each excluded residue $r \bmod 30$, put $c_r = \mathrm{spf}(\gcd(r,30)) - 1$ and $g_r = \min\{g \ge 1 : \gcd(r+g,30) = 1\}$. Convexity of $c_r\{(30x+r)^{-1} - (30x+r+g_r)^{-1}\}$ bounds the terms beyond the finite prefix, giving

$$\begin{aligned}\delta_0 \le \bar{\delta}_0 := &\sum_{p=17}^{299} (\mathrm{spf}(p) - 1)\left(\frac{1}{p} - \frac{1}{q(p)}\right) \\ &+ \sum_{\substack{0 \le r < 30 \\ \gcd(r,30) > 1}} \frac{c_r}{30}\left(y_r - \frac{y_r^2}{2} + \frac{y_r^3}{3}\right), \qquad y_r = \frac{g_r}{285 + r}.\end{aligned} \tag{A.18}$$

This uses the midpoint integral from $19/2$ and $\log(1+y) \le y - y^2/2 + y^3/3$.

Group-loss bounds. For a near group, consider each possible support size $1 \le d \le |\mathcal{P}_j|$. The function $\mathcal{M}_m$ is convex and satisfies $\mathcal{M}_m(-s) = \mathcal{M}_m(s) + s$. For $s \ge 0$, its maximum over the Bernoulli parameter occurs at zero, one, or a valid point $p = (k+1-s)/(m+1)$ with $k \le mp + s \le k+1$. Indeed, differentiation on each such interval gives

$$m\{k + 1 - s - (m+1)p\} \Pr\{\mathrm{Bin}(m-1, p) = k\}.$$

All candidates are rational. The remaining integral can therefore be bounded by a finite calculation. Convexity gives

$$\Theta_m(f) \le \widehat{\Theta}_m(f) := \frac{1}{16}\left\{\frac{\mathcal{M}_m(f-1) + \mathcal{M}_m(f+1)}{2} + \sum_{k=1}^{15} \mathcal{M}_m(f - 1 + k/8)\right\}. \tag{A.19}$$

Consequently the near-group deletion charge is bounded by $\sum_{j=0}^{2} \max_{1 \le d \le |\mathcal{P}_j|} \widehat{\Theta}_{d-1}(f_{j,d})$.

For the far groups put $P = 449 \cdot 3^j$ and $d_j = |\mathcal{A} \cap [P, 3P)|$. Uniform bounds for the rate and premature-service charges are, respectively,

$$r_j^+ = \frac{\Gamma^2}{P}\{\sqrt{(d_j+3)b_j} + P^{-4}\}, \qquad e_j^+ = \frac{\sqrt{d_j+3}}{4e\sqrt{b_j}} e^{-2b_j}.$$

Here a group with fewer occupied periods is bounded by the same expressions, and a deleted group incurs only its rate charge. Exact residue counting gives $d_j + 3 \le (8/15)P + 5$. Evaluate a finite prefix $j < J$ with the exact counts. From J onward, replace $d_j + 3$ by this last bound. The square-root rate term has successive ratio at most $\sqrt{(b_J + \kappa)/(3b_J)}$, its P^{-5} remainder has ratio 3^{-5}, and the premature-service term has ratio at most $\sqrt{3}e^{-2\kappa}$. For $b_* > \kappa/2$ and $\sqrt{3}e^{-2\kappa} < 1$, these ratios are below one, giving explicit geometric upper bounds on both infinite tails.

These bounds can be certified using rational arithmetic. Square roots are enclosed by rational squaring, and

$$e^{-x} \le \left(\sum_{k=0}^{m_{\exp}} \frac{x^k}{k!}\right)^{-1} \qquad (x \ge 0)$$

provides rational upper bounds for the exponential terms.

The preceding calculations give the bounds R_+ and E_+ by rationally enclosing, respectively,

$$\begin{aligned}R_{\mathrm{tot}} &= \bar{\delta}_0 + \sum_{j=0}^{2} \Gamma(c_j + P_j^{-5}) + \sum_{j \ge 0} r_j^+, \\ E_{\mathrm{tot}} &= \sum_{j=0}^{2} \max_{1 \le d \le |\mathcal{P}_j|} \widehat{\Theta}_{d-1}(f_{j,d}) + \sum_{j \ge 0} e_j^+.\end{aligned} \tag{A.20}$$

Joint head and rate loss. For the head laws in Appendix A.1, replace the free Bernoulli sum by a binomial using the same convex-order argument as above, and write $N = \mathrm{Bin}(m,p) + Z_H$. For each of the $13 \cdot 7$ head package-law pairs, Z_H contains its fixed non-Bernoulli components; any

Bernoulli package is included in the binomial term. Put $\alpha = \mathbb{E}Z_H$. On $k \le mp + \alpha + s \le k + 1$, the derivative of the stop loss is

$$m\{k + 1 - \alpha - s - (m + 1)p\} \Pr\{\mathrm{Bin}(m - 1, p) + Z_H = k\}.$$

Its maximum is therefore attained at zero, one, or a valid rational point $p = (k+1-\alpha-s)/(m+1)$. Let $\mathcal{F}(s)$ be the largest of these finitely evaluated values over all the head laws. Then $F_N(s) \le \mathcal{F}(s)$ for every head arising from the construction.

A fixed numerical certificate. Use the following fixed parameters:

$$\Gamma = \frac{2001}{2000}, \quad (c_0, c_1, c_2) = \left(\frac{171}{1000}, \frac{1}{10}, \frac{3}{40}\right), \quad \xi = 10^{-6}, \quad \nu = 10^{-4}, \quad (b_*, \kappa) = \left(\frac{14}{5}, \frac{11}{20}\right).$$

For each far group, round $\sqrt{(d_g + 3)b_j}$ upward on $(4P^4)^{-1}\mathbb{Z}$ to obtain f. Its error is at most $e_P/4$; the displayed value of Γ then satisfies the enclosure condition used in (A.12). For every retained group, rational bisection chooses δ according to (A.9).

Evaluate (A.20) with $J = 24$, square-root enclosures of width 10^{-24}, and $m_{\exp} = 219$. The rational comparisons just described, reproduced by `proof-checks/verify-constants.py`, give

$$\bar{s} + \mathcal{F}(\bar{s}) < \frac{93}{100}, \qquad E_+ < \frac{1}{25}, \qquad c_{\mathrm{loss}} < \frac{97}{100}.$$

Computational complexity. Store an aggregated stream as a tree: leaves label arms, and each internal node cycles through the streams combined at that aggregation step. For fixed numerical parameters, the main construction sizes are

$$\#\text{LP entries} = O(nQ), \quad \#\text{tree nodes} = O(n), \quad \text{quota-window length} < 2Q + h.$$

The trees have disjoint leaf sets and depth $O(\log Q)$. A sieve gives prime factors; integer division and remainders return arm labels at positive or negative request indices, and floor counts locate preceding same-arm requests. A node's repeating sequence has length equal to its child count times the least common multiple of the child sequence lengths. Sequence positions have polynomial bit length, so phases can be sampled and requested slots generated using the tree representation.

The near-group constants also admit a finite computation. For a candidate η, put

$$c_p(\eta) = \max_{1 \le r \le p-1} r(1/p^2 - \eta/p).$$

Choose $r = p - 1$ if its coefficient is nonnegative and $r = 1$ otherwise. The sum of the d largest $c_p(\eta)$ is nonpositive exactly when $\eta \ge h_{j,d}$. A zero sum and its selected multiplicities certify the optimum. The piecewise-linear endpoints and pairwise crossings give a polynomial finite search for this zero.

The group-deletion test bounds the separation of the buffer target from W, so doubling and rational bisection take polynomially many bit operations. Sequence positions, phases, rates, and the quota origin have polynomial bit length. Together with the flow-rounding update bound, this makes each candidate construction and score evaluation polynomial in the encoded input and output length.

B Proof of Theorem 2

Proof of Theorem 2. For $K = L = 1$, we construct a three-arm instance with renewal value $8/9$ and optimal long-run reward $3/4$. Let

$$R_A(d) = \begin{cases} 0 & d = 1, \\ 3/4 & 2 \le d \le 4, \\ 1 & d \ge 5, \end{cases} \qquad R_B(d) = \mathbf{1}\{d \ge 3\}, \qquad R_C(d) = \mathbf{1}\{d \ge 5\}.$$

The following primal choice has load one and value $8/9$:

$$x_{A,2} = 8/9, \quad x_{A,5} = 1/9, \quad x_{B,3} = x_{C,5} = 1$$

For the dual, the choice

$$\lambda = 7/12, \qquad (\nu_A, \nu_B, \nu_C) = (1/12, 5/36, 1/12)$$

satisfies $R_i(d) \le \lambda + \nu_i d$ for every integer d. Consequently

$$\frac{8}{9} \le \mathrm{UB} \le \lambda + \nu_A + \nu_B + \nu_C = \frac{8}{9}.$$

The repeating word (A, B, A, C, A, B, A, I), where I denotes idling, earns $3/4$ per date.

To prove optimality, we count the space left for A between rewarding B/C services. Remove zero-reward services, which only increases subsequent ages, and first consider a cyclic calendar of length T. Let the M positive B/C services have successive circular gaps $g_1, \dots, g_M$, summing to T. There cannot be three consecutive gaps at most two: four B/C services would have to alternate, forcing two C services within four dates. If $M = 0$, A alone earns at most $3/8$ per date, so assume $M > 0$.

Call a gap large if it is at least three, and put

$$X = \sum_j (\lceil g_j/2 \rceil - 1), \qquad S = 3X - M.$$

Let Q_0 count unit gaps with a large gap on both sides. If there are b large gaps, their successive separations s_ℓ in the gap sequence satisfy

$$s_\ell \in \{1, 2, 3\}, \qquad \sum_{\ell=1}^{b} s_\ell = M.$$

Each gap counted by Q_0 has separation two. Therefore

$$Q_0 \le \sum_{\ell=1}^{b} (3 - s_\ell) = 3b - M \le 3X - M = S.$$

The free segment between two B/C services has $g_j - 1$ dates and can hold at most $\lfloor g_j/2 \rfloor$ A services. If it contains m_j, define $D_j = \lfloor g_j/2 \rfloor - m_j$ and $D_0 = \sum_j D_j$. Then

$$N_A = T - \sum_j \lceil g_j/2 \rceil - D_0.$$

An A service normally earns $3/4$; it earns an extra $1/4$ only after a gap of at least five. Such a long gap must either use extra empty space or pass through a tightly placed pair of B/C services. Let N_{long} count these long A gaps. We claim

$$N_{\mathrm{long}} \le D_0 + Q_0.$$

In a nonempty free segment of length ℓ with $m \ge 1$ A services, let b_0, b_m be its empty endpoint runs and $b_i \ge 1$ its internal empty runs. Their excess-space total is

$$b_0 + b_m + \sum_{i=1}^{m-1} (b_i - 1) = \ell - (2m - 1) \le 2D_j + 1.$$

Call an empty endpoint run of length at least two an endpoint defect, and an internal A gap of at least five an internal defect. They use at least two and three units of excess space, respectively. Since the total is at most $2D_j + 1$, there are at most D_j such defects.

A nonempty segment containing no A has $D_j \ge 1$ and can be crossed by at most one consecutive A gap. Charge each long A gap to an internal defect, a wholly crossed empty segment, or an endpoint defect, when available. No defect or empty segment is charged twice. If none applies, the A gap crosses exactly two adjacent B/C services, one date apart, with its two endpoint distances both two. It then corresponds to a unit gap counted in Q_0. Thus $N_{\mathrm{long}} \le D_0 + Q_0$.

The total reward consequently satisfies

$$\begin{aligned} R &= \tfrac{3}{4} N_A + M + \tfrac{1}{4} N_{\mathrm{long}} \\ &= \tfrac{3}{4} T - \tfrac{1}{4} S - \tfrac{3}{4} D_0 + \tfrac{1}{4} N_{\mathrm{long}} \\ &\le \tfrac{3}{4} T - \tfrac{1}{4} S - \tfrac{1}{2} D_0 + \tfrac{1}{4} Q_0 \le \tfrac{3}{4} T. \end{aligned}$$

Appending five idle dates to a finite calendar and repeating it makes all boundary rewards no smaller, since the curves saturate by age five. The cyclic bound therefore upper-bounds arbitrary finite calendars by $3T/4 + O(1)$, and also bounds adaptive and randomized long-run policies. Thus $\mathrm{OPT} = 3/4$, giving gap $5/36$.

Scale the three-arm instance by L. For general K, add $K-1$ arms with constant reward L. These can be served every date without loss: replace any nonconstant activation if necessary, earning at least as much immediately, while the deleted service only increases that arm's future age. Both the relaxation and the optimal hard-cap value increase by $(K-1)L$. The gap stays $5L/36$, with $n = K+2$, completing the theorem. □

C Proof of Theorem 3

We prove the expected-reward guarantee by separating the loss of each MBI calendar from the error in its optimistic reward curves. The latter is charged at the waits actually realized during execution. Their total duration bounds the accumulated learning loss.

Write $\overline{S}_T = \sum_{t\le T}\sum_{i\in A_t} R_i(d_i(t))$ for the conditional-mean reward along the realized action path. The model gives $\mathbb{E}S_T = \mathbb{E}\overline{S}_T$. We use the computable score S_{poly} from (A.16), which satisfies $S_{\mathrm{poly}}(R) \ge \mathrm{UB}(R) - L$.

C.1 Retained observations and confidence curves

Epoch j comprises rounds $T_j, \dots, 2T_j - 1$, where $T_j = 2^j$. An observation at round t and actual wait d is retained precisely when $d \le 2^{\lfloor \log_2 t \rfloor}$, and enters the estimates after its block. This eligibility decision and the block calendar are fixed before the corresponding rewards are observed. Retained counts, reward sums, and physical ages carry over across epochs.

Fix $\delta = 1/20$ throughout the proof and Algorithm C.1. The parameter choices in (C.10) satisfy

$$2nM_j e^{-\beta_j} \le \delta_j, \qquad \beta_j \le \ell_j = O(\log(2nT_j)), \qquad \sum_{j\ge 0} T_j\delta_j = O(1).$$

Here M_j counts possible wait–sample-count pairs for one arm by the end of epoch j:

$$M_j = \sum_{d=1}^{T_j}\left(1 + \left\lfloor \frac{2T_j - 2}{d} \right\rfloor\right).$$

The first observation at a wait may use the initial age; subsequent observations at that wait are separated by at least d rounds.

For a normalized empirical mean m from $N > 0$ retained observations, let $q_\beta(m, N)$ be the exact KL-UCB upper endpoint and $b_\beta(N)$ the radius $\sqrt{\beta/(2N)}$ rounded upward with error at most T_j^{-2}, as defined in (C.13). The implementation returns an endpoint satisfying

$$q_{\beta_j}(m, N) \le \widetilde{u} \le \min\{1, m + b_{\beta_j}(N)\}. \tag{C.1}$$

It returns one when $N = 0$ or $m = 1$. At each block boundary, set $u_i(d) = L\widetilde{u}$ from the retained observations at wait d, and form

$$U_i(d) = \min_{d\le q\le T_j} u_i(q) \quad (d \le T_j), \qquad U_i(d) = L \quad (d > T_j). \tag{C.2}$$

Monotonicity makes an upper bound at a longer wait valid at a shorter wait, so the suffix minimum gives MBI nondecreasing upper curves.

Lemma C.1 (Retained KL confidence)**.** *There are events G_j with $\Pr(G_j^c) \le \delta_j$ such that, at every block start in epoch j, $R_i(d) \le U_i(d)$ for all waits and*

$$U_i(d) - R_i(d) \le 2b_{i,d,e} \quad (d \le T_j), \qquad b_{i,d,e} = \begin{cases} L, & N_{i,d,e} = 0, \\ Lb_{\ell_j}(N_{i,d,e}), & N_{i,d,e} > 0. \end{cases} \tag{C.3}$$

Here $N_{i,d,e}$ is the retained sample count at the start of block e.

Proof. We apply the KL confidence construction (Garivier & Cappé, 2011; Cappé et al., 2013; Maillard et al., 2011) to adaptive visits under the conditional bounded-observation model. Fix an arm–wait pair with mean $\mu = R_i(d)/L$. For every real λ, convexity gives $\mathbb{E}[e^{\lambda X} \mid \text{past}] \le 1-\mu+\mu e^{\lambda}$ for its normalized observation X. Along eligible visits this produces an exponential supermartingale. Stopping at the Nth visit, with truncation if that visit never occurs, and optimizing the Chernoff parameter gives

$$\Pr\{\mu > q_{\beta_j}(m, N),\ N \text{ visits occur}\} \le e^{-\beta_j}.$$

The bounded-variable exponential bound of Hoeffding (1963) applied at the same stopping time gives

$$\Pr\{m - \mu > \sqrt{\beta_j/(2N)},\ N \text{ visits occur}\} \le e^{-\beta_j}.$$

The two tails control coverage and the width of the upper endpoint. A union bound over the nM_j state–count pairs yields

$$\Pr(G_j^c) \le 2nM_j e^{-\beta_j} \le \delta_j.$$

Every block-start estimate uses one of these pairs. On G_j, (C.1) gives coverage and width at most $2Lb_{\beta_j}(N) \le 2b_{i,d,e}$. Zero counts obey the same bound. For $q \ge d$, monotonicity gives $R_i(d) \le R_i(q) \le u_i(q)$. Taking the suffix minimum preserves coverage and bounds the width by the upper endpoint at d itself. The tail value L covers $d > T_j$. □

C.2 Blockwise planning loss

Each block freezes U and executes a calendar at the current physical ages a. The offline certificate (A.17) gives

$$J_U(C_e; a) \ge h_e S_{\text{poly}}(U) - nL. \tag{C.4}$$

The allocation and relabeling rules in Appendix C.4 preserve this certificate.

Lemma C.2 (Blockwise loss decomposition)**.** *On G_j, a completed block e satisfies*

$$h_e S_{\text{poly}}(R) - J_R(C_e; a) \le 2nL + 2\sum_{i,d} C_{i,d,e} b_{i,d,e}, \tag{C.5}$$

where $C_{i,d,e}$ counts pulls of arm i at actual wait d, excluding its first pull in the block. For each arm these counts satisfy

$$\sum_{e,d} dC_{i,d,e} \le T_j, \qquad dC_{i,d,e} \le h_j, \qquad N_{i,d,e} \ge \sum_{f<e} C_{i,d,f}. \tag{C.6}$$

The sums range over completed blocks of epoch j.

Proof. The LP value and S_{poly} are coordinatewise nondecreasing, so $S_{\text{poly}}(U) \ge S_{\text{poly}}(R)$. Add and subtract the optimistic reward of the same calendar:

$$\begin{aligned} h_e S_{\text{poly}}(R) - J_R(C_e; a) &\le h_e S_{\text{poly}}(U) - J_R(C_e; a) \\ &= \underbrace{h_e S_{\text{poly}}(U) - J_U(C_e; a)}_{\text{planning loss}} + \underbrace{J_U(C_e; a) - J_R(C_e; a)}_{\text{estimation error}}. \end{aligned}$$

The first term is at most nL. In the second term, each arm's first pull contributes at most L. Every subsequent wait has both endpoints inside the block, hence $d \le h_e \le T_j$; Lemma C.1 therefore bounds its contribution by $2b_{i,d,e}$. This proves (C.5).

For one arm, these completed waiting intervals are disjoint and lie inside the epoch. Their total length gives the first two bounds in (C.6). Each corresponding observation is eligible and committed before the next block, giving the last bound. Earlier epochs and first pulls can further increase the retained count. □

C.3 Accumulating the learning loss

We sum the confidence widths at realized waits using the waiting-time budget. Fix (i, d) and write $c_e = C_{i,d,e}$, $x_e = \sum_{f<e} c_f$, and $m = \lfloor h_j/d \rfloor$. For $m = 0$ the count is zero. Otherwise $c_e \le m$, and blocks starting with $x_e < m$ contain fewer than $2m$ such pulls in total. For the later blocks, $c_e \le m \le x_e$, so

$$\frac{c_e}{\sqrt{x_e}} \le \frac{1+\sqrt{2}}{2} \int_{x_e}^{x_e+c_e} u^{-1/2}\, du.$$

Summing bounds the cost of using counts frozen at block starts:

$$\sum_e \frac{c_e}{\sqrt{\max\{1, x_e\}}} \le 2m + (1+\sqrt{2})\sqrt{\sum_e c_e}.$$

Long waits occur less often. Applying weighted Cauchy–Schwarz to (C.6), with $\mathcal{H}_s = \sum_{d=1}^{s} 1/d$, gives

$$\sum_d \sqrt{\sum_e C_{i,d,e}} \le \sqrt{\left(\sum_{e,d} d C_{i,d,e}\right)\left(\sum_{d=1}^{T_j} \frac{1}{d}\right)} \le \sqrt{T_j \mathcal{H}_{T_j}}.$$

Together with $b_{i,d,e} \le L\sqrt{2\ell_j}/\sqrt{\max\{1, x_e\}} + L/T_j^2$, this yields

$$\sum_{e,i,d} C_{i,d,e} b_{i,d,e} \le L\sqrt{2\ell_j}\{2nh_j\mathcal{H}_{T_j} + (1+\sqrt{2})n\sqrt{T_j\mathcal{H}_{T_j}}\} + KL/T_j. \tag{C.7}$$

For a prefix of epoch j, completed blocks contribute their losses from (C.5); the unfinished block contributes at most KLh_j, since $S_{\mathrm{poly}} \le KL$ and rewards are nonnegative. Thus, on G_j, an epoch prefix of length $t \le T_j$ obeys

$$tS_{\mathrm{poly}}(R) - \overline{S}_{j,t} \le O\left(nL\left\{\frac{T_j}{h_j} + h_j\mathcal{H}_{T_j}\sqrt{\ell_j} + \sqrt{T_j\mathcal{H}_{T_j}\ell_j}\right\}\right). \tag{C.8}$$

Here $\overline{S}_{j,t}$ is its conditional-mean reward, and the hidden constant is absolute. We used $K \le n$ to absorb the unfinished-block and numerical-rounding terms. The first term is the cost of restarting calendars; the second is the cost of delaying estimate updates.

The block scale in (C.11) balances these two terms: up to a fixed factor and clipping to $[1, T_j]$, it is $\sqrt{T_j/(\mathcal{H}_{T_j}\sqrt{\ell_j})}$. Substitution into (C.8) gives

$$tS_{\mathrm{poly}}(R) - \overline{S}_{j,t} \le O\left(nL\left\{\sqrt{T_j\mathcal{H}_{T_j}\ell_j} + \mathcal{H}_{T_j}\sqrt{\ell_j}\right\}\right) \quad \text{on } G_j.$$

The second term covers epochs where the block scale is clipped at one.

Taking expectations epoch by epoch adds at most $KL\sum_j T_j\delta_j = O(KL)$ for confidence failures. Let $\ell_T^* = \max_{j:T_j\le T}\ell_j$. Using $\sum_{j:T_j\le T}\sqrt{T_j} = O(\sqrt{T})$ and $O(\log(2T))$ epochs, and absorbing $O(KL)$ using $K \le n$, gives

$$TS_{\mathrm{poly}}(R) - \mathbb{E}S_T \le O\left(nL\left\{\sqrt{T\mathcal{H}_T\ell_T^*} + \mathcal{H}_T\sqrt{\ell_T^*}\log(2T)\right\}\right). \tag{C.9}$$

Since $\mathcal{H}_T = O(\log(2T))$, $\ell_T^* = O(\log(2nT))$, and $S_{\mathrm{poly}}(R) \ge \mathrm{UB}(R) - L$, dividing (C.9) by T proves the reward guarantee in Theorem 3.

C.4 Implementation and computation

The following allocation and relabeling rules choose among plans while preserving the block certificate (C.4). We also give the fixed parameters and rational KL-UCB implementation.

Parameter choices. With M_j as defined in Appendix C.1, choose

$$A_j = \frac{5nM_jT_j(j+1)^2}{2\delta}, \qquad \delta_j = \frac{4\delta}{5T_j(j+1)^2},$$
$$\ell_j = \left\lceil \log_2 \frac{16nT_j^4}{3\delta} \right\rceil, \qquad \ln A_j \le \beta_j \le \ell_j, \tag{C.10}$$

where β_j is rational. These choices give $2nM_je^{-\beta_j} \le \delta_j$ and $\sum_{j\ge0} T_j\delta_j = (4\delta/5)\sum_{j\ge0}(j+1)^{-2} = O(1)$. The exact execution length is

$$\underline{h}_j = \min\left\{T_j, \max\left\{1, \left\lceil \sqrt{\frac{T_j}{\mathcal{H}_{T_j}\sqrt{\ell_j}}} \right\rceil\right\}\right\}, \qquad h_j = \min\{T_j, 8\underline{h}_j\}. \tag{C.11}$$

Blocks are shortened at epoch boundaries.

Selecting an optimal allocation. On optimistic plateaus, BALANCEDBASIC spreads service across arms while preserving LP optimality. Let $Q = 100n$ and $I = \{i : U_i(p) = L \text{ for some } p \le Q\}$. Set

$$r_i = \min\{p \le Q : U_i(p) = L\} \quad (i \in I), \qquad f(q) = \sum_{i\in I} \frac{1}{\max\{r_i, q\}}.$$

If $f(1) < K$, BALANCEDBASIC returns an optimal basic solution of the finite LP. Otherwise choose the smallest integer q_* with $f(q_*) \le K$, set $p_i = \max\{r_i, q_*\}$ for $i \in I$, and let the other arms idle. Fill the remaining load by shortening eligible q_* periods to $q_* - 1$ in arm order. Each full shortening adds $\Delta = 1/(q_* - 1) - 1/q_*$. If the remaining load is $b < \Delta$, assign time shares b/Δ and $1 - b/\Delta$ to these two periods and stop.

Lemma C.3 (Balanced optimal solution)**.** *For nondecreasing curves in $[0, L]$ and $Q \ge n$,* BALANCEDBASIC *returns an optimal basic solution of the finite renewal LP. When $f(1) \ge K$, every arm in I is active, at most one arm mixes two adjacent periods, and the objective equals KL.*

Proof. In this branch, $|I| \ge K$ and $f(\lceil |I|/K \rceil) \le K$, so $q_* \le n \le Q$. If $f(q_*) = K$, the initial allocation fills capacity. Otherwise $q_* \ge 2$ and $f(q_* - 1) > K$, so the sequential shortenings fill capacity with at most one fractional choice. All selected periods are at least r_i, and therefore have reward L. The objective is KL, matching the upper bound of L times total load.

After adding an idle variable for each arm, each arm has one positive variable except possibly one with two. The arm equalities fix all single-variable weights. The exceptional arm's equality and capacity equality fix its two weights, since its periods differ. Thus the positive support columns are linearly independent. The ordinary-solver branch returns a basic optimum. □

Continuing service across blocks. MBI uses the selected basic solution in its calendar construction. For arms assigned a single common period, CONTINUE assigns earlier-starting service sequences to older arms, subject to the nominal-reward certificate. If s_i is arm i's first service position in the block, set $s_i = h + 1$ when it is absent. Sort sequences by $(s_i, -a_i, i)$ and arms by $(-a_i, i)$, and pair them in these orders. Idle and exceptional arms keep their assignments. Let C' be the relabelled calendar and Z the score returned by MBI. Set

$$\text{CONTINUE}(C, x^*, a, U, Z) = \begin{cases} C', & J_U(C'; a) \ge Z, \\ C, & \text{otherwise.} \end{cases} \tag{C.12}$$

Relabeling within a period class preserves per-round capacity and distinctness. The acceptance test uses only U and the inherited ages. Since $J_U(C; a) \ge Z \ge hS_{\text{poly}}(U) - nL$, the accepted calendar satisfies (C.4). In Algorithm C.1, $W_i(d)$ denotes the retained reward sum, so the normalized empirical mean is $W_i(d)/(LN_i(d))$ when $N_i(d) > 0$.

Algorithm C.1 Detailed implementation of MBI-KL

Input: n, K, L, initial ages a.

$t \leftarrow 1$; $N_i(d) \leftarrow 0$, $W_i(d) \leftarrow 0$ for all i, d.
for $j = 0, 1, \ldots$ **do**
  $T_j \leftarrow 2^j$; choose (β_j, h_j) via (C.10)–(C.11).
  while $t < 2T_j$ **do**
    $h \leftarrow \min\{h_j, 2T_j - t\}$; $\Delta N \leftarrow 0$, $\Delta W \leftarrow 0$.
    Form U from (N, W) using (C.1)–(C.2).
    $x^* \leftarrow \textsc{BalancedBasic}(U, K, 100n)$.
    $(C, Z) \leftarrow \text{MBI}(U, K, L, a, h; x^*)$.
    $C \leftarrow \textsc{Continue}(C, x^*, a, U, Z)$.
    for $\tau = t, \ldots, t + h - 1$ **do**
      $A_\tau \leftarrow C_{\tau - t + 1}$; $E_\tau \leftarrow \{i \in A_\tau : a_i \le T_j\}$.
      Pull A_τ and observe $(Y_{i,\tau})_{i \in A_\tau}$.
      $(\Delta N_i(a_i), \Delta W_i(a_i)) \leftarrow (\Delta N_i(a_i) + 1, \Delta W_i(a_i) + Y_{i,\tau})$ for $i \in E_\tau$.
      $a_i \leftarrow 1$ if $i \in A_\tau$, and $a_i \leftarrow a_i + 1$ otherwise.
    end for
    $N \leftarrow N + \Delta N$; $W \leftarrow W + \Delta W$; $t \leftarrow t + h$.
  end while
end for

Rational confidence computation. The exact upper endpoint and rounded radius used above are

$$q_\beta(m, N) = \sup\{q \in [m, 1] : N \,\mathrm{kl}(m, q) \le \beta\}, \qquad b_\beta(N) = \frac{1}{T_j^2} \left\lceil T_j^2 \sqrt{\frac{\beta}{2N}} \right\rceil. \tag{C.13}$$

For rational data, compute the endpoint using interval arithmetic. By monotonicity of KL, a candidate $q > m$ is above its exact root whenever a lower enclosure certifies $N \,\mathrm{kl}(m, q) \ge \beta_j$; $q = 1$ is also valid. Pinsker's inequality (Lattimore & Szepesvári, 2019, Lemma 10.2) gives $q_{\beta_j}(m, N) \le m + \sqrt{\beta_j/(2N)}$. Capping the candidate at $\min\{1, m + b_{\beta_j}(N)\}$ therefore proves (C.1), with radius rounding error at most T_j^{-2}.

The implementation rounds an upper enclosure of $\ln A_j$ upward to $2^{-32}\mathbb{Z}$ to obtain β_j, and checks at most eight bisection proposals on $T_j^{-2}\mathbb{Z}$, using $q = 1$ if none is certified. The bounds $M_j \le 2T_j^2$ and $(j+1)^2/T_j \le 9/4$ leave a constant margin between $\ln A_j$ and ℓ_j, so logarithm enclosures of polynomial encoded length suffice for the prescribed rounding.

Running time. The ceiling in (C.11) is found by exact integer comparisons with $h^4 \mathcal{H}_{T_j}^2 \ell_j \ge T_j^2$. The rational optimistic curves have finite descriptions: retain the raw bounds below L and the first terminal value L. Exact plateau compression preserves the finite LP, which has $O(nQ)$ possible variables. The balancing rule searches $q_* \le n$ using rational load increments, and continuation takes at most $O(n \log n)$ comparisons plus calendar relabeling and score evaluation. MBI has the expected output-polynomial complexity proved in Appendix A. Counts, observations, confidence endpoints, and log enclosures have polynomial encoded length in the data and elapsed rounds. Summing over block calls gives the expected polynomial running time in Theorem 3.

D Proof of Theorem 4

We hide a particularly rewarding waiting time for each arm. The proof compares a renewal plan that knows these times with the number of times a learner can discover and use them. Waiting limits both observations and reward collection.

Normalize $L = 1$ by dividing rewards by L. Let G be a sufficiently large integer and set

$$K = 16G, \qquad n = 16G^2, \qquad T = G^4. \tag{D.1}$$

Fix any randomized online policy. All arms start at age one; for counting completed waiting intervals, we represent the initial age by a virtual activation at date zero.

Let

$$a = \lceil 3G/8 \rceil, \qquad \mathcal{J} = \{a, \ldots, 2G - a\}, \qquad m = |\mathcal{J}|, \qquad \epsilon = \frac{1}{2G}.$$

Independently for each arm, draw a hidden age with probabilities

$$\Pr(\theta_i = j) = \nu_j = \frac{j}{Gm}, \quad j \in \mathcal{J}, \qquad R_i^\theta(d) = \epsilon\big\{\min(d, 2G) + \mathbf{1}\{d = \theta_i\}\big\}. \tag{D.2}$$

Since $\mathcal{J}$ is symmetric about G, the probabilities sum to one and $\mathbb{E}(1/\theta_i) = 1/G$. This matches the informed plan's expected capacity demand. Weighting ages by j also aligns the prior with the j rounds consumed by each test, as used in the discovery bound below. The linear baseline offsets the disappearance of the one-age bonus, keeping the curves nondecreasing. Conditional on θ, each activation returns an independent Bernoulli observation with its prescribed mean. Expectations with subscript θ average over the hidden ages as well as feedback and policy randomization.

Lemma D.1 (Valid curves and waiting budgets)**.** *The curves in (D.2) are nondecreasing and bounded by one. Let $N_{i,j}(t)$ count completed age-j activations of arm i by date t, including the first activation with its gap measured from date zero. Every action path satisfies*

$$\sum_{j \in \mathcal{J}} j N_{i,j}(t) \le t, \qquad \overline{S}_T \le n\epsilon T + \epsilon \sum_i N_{i,\theta_i}(T), \tag{D.3}$$

where $\overline{S}_T = \sum_{t \le T} \sum_{i \in A_t} R_i^\theta(d_i(t))$.

Proof. Before saturation, each increment is ϵ, except when entering the hidden age, where it is 2ϵ, and leaving it, where it is zero. The hidden age is below $2G$, so the curve never exceeds one.

Completed gaps of the same arm do not overlap. Their total length is at most t, giving the first inequality in (D.3). For completed gaps $d_{i,1}, d_{i,2}, \ldots$, the baseline reward obeys

$$\epsilon \sum_k \min(d_{i,k}, 2G) \le \epsilon \sum_k d_{i,k} \le \epsilon T.$$

Only exact hits on θ_i earn the bonus. Summing over arms gives the second inequality. The conditional-mean assumption also gives $\mathbb{E}S_T = \mathbb{E}\overline{S}_T$. □

When the informed plan's demand exceeds capacity, mix its periods with $2G$. This preserves the baseline reward while reducing demand and the hidden bonus.

Lemma D.2 (A feasible renewal-LP comparison)**.** *Let $v = \mathrm{Var}(G/\theta_i)$. Then*

$$\mathbb{E}_\theta \mathrm{UB}(R^\theta) \ge n\epsilon + \epsilon\left(K - \sqrt{Kv/G}\right). \tag{D.4}$$

Proof. Write $Q = \sum_i 1/\theta_i$. Independence and $n = KG$ give

$$\mathbb{E}Q = K, \qquad \mathrm{Var}(Q) = Kv/G.$$

If $Q \le K$, use the hidden period for each arm. If $Q > K$, allocate time fraction γ to that period and the rest to $2G$, where

$$\gamma = \frac{K/2}{Q - K/2}, \qquad \gamma Q + (1 - \gamma)\frac{n}{2G} = K.$$

This is a feasible time-fraction solution of the renewal LP. Each arm contributes baseline ϵ per date under either period, so

$$\mathrm{UB}(R^\theta) \ge n\epsilon + \epsilon B(Q), \qquad B(Q) = \begin{cases} Q, & Q \le K, \\ KQ/(2Q - K), & Q > K. \end{cases}$$

For $Q \le K$, $B(Q) = K - |Q - K|$. For $Q > K$,

$$B(Q) - (2K - Q) = \frac{2(Q - K)^2}{2Q - K} \ge 0.$$

Thus $B(Q) \ge K - |Q - K|$ in both cases. Taking expectations and using $\mathbb{E}|Q - K| \le \sqrt{\mathrm{Var}(Q)}$ proves the claim. □

To bound the learner's exact-age hits, apply the adaptive KL decomposition and Pinsker inequality (Lattimore & Szepesvári, 2019, Section 14.3 and Lemma 15.1) at each time prefix. Together with the waiting budget, they bound the number of hidden-age discoveries.

For $j \in \mathcal{J}$, define

$$q_j = \frac{j}{2G}, \qquad w_j = \min\{q_j(1-q_j), (q_j+\epsilon)(1-q_j-\epsilon)\}, \qquad b_j = \frac{1}{2\sqrt{w_j}},$$

and write $W_1 = \sum_j b_j^2/j$ and $W_{-1} = \sum_j j b_j^2$.

Lemma D.3 (Armwise adaptive discovery)**.** *For each arm i, every online policy satisfies*

$$\mathbb{E}_\theta N_{i,\theta_i}(T) \le \frac{T}{Gm} + \frac{\epsilon}{Gm}\left\{\frac{2}{3}T^{3/2}\sqrt{W_1} + 2\sqrt{T}\sqrt{W_{-1}}\right\}. \tag{D.5}$$

Proof. Fix i and the other hidden ages θ_{-i}. Let P_0 be the law of the entire adaptive trajectory when only arm i's bonus is removed, and let P_j put that bonus at age j. Set

$$A_j(t) = \mathbb{E}_0 N_{i,j}(t), \qquad \sum_j j A_j(t) \le t.$$

The observation laws differ only when arm i is activated at age j. For such an observation, the two means are q_j and $q_j + \epsilon$. The Bernoulli divergence satisfies

$$\mathrm{kl}(q_j, q_j+\epsilon) = \int_{q_j}^{q_j+\epsilon} \frac{u-q_j}{u(1-u)}\,du \le \frac{\epsilon^2}{2w_j}.$$

Consequently, the adaptive chain rule yields, at every integer t,

$$\mathrm{KL}(P_0^{\le t} \| P_j^{\le t}) \le \frac{\epsilon^2}{2w_j} A_j(t). \tag{D.6}$$

Partition the dates into consecutive blocks of length j, allowing a short final block. Two age-j activations of the same arm are at least j dates apart. Hence a block's hit count $X_{j,b}$ is zero or one. If the block ends at s_b, Pinsker gives

$$\mathbb{E}_j X_{j,b} \le \mathbb{E}_0 X_{j,b} + \epsilon b_j \sqrt{A_j(s_b)}. \tag{D.7}$$

For any nonnegative nondecreasing function f, these block endpoints obey

$$\sum_b f(s_b) \le \frac{1}{j}\int_0^T f(t)\,dt + 2f(T).$$

Indeed, the upper and lower Riemann sums on complete blocks differ by at most $f(T)$, and the final short block contributes at most $f(T)$. Apply this with $f(t) = \sqrt{A_j(\lfloor t \rfloor)}$ to obtain

$$\mathbb{E}_j N_{i,j}(T) \le A_j(T) + \frac{\epsilon b_j}{j}\int_0^T \sqrt{A_j(t)}\,dt + 2\epsilon b_j\sqrt{A_j(T)},$$

where A_j is extended as a step function.

We now average over the hidden age. The null-law term satisfies

$$\sum_j \nu_j A_j(T) = \frac{1}{Gm}\sum_j j A_j(T) \le \frac{T}{Gm}.$$

Weighted Cauchy–Schwarz and the same waiting budget give

$$\sum_j \frac{\nu_j b_j}{j}\sqrt{A_j(t)} \le \frac{\sqrt{tW_1}}{Gm}, \qquad \sum_j \nu_j b_j \sqrt{A_j(T)} \le \frac{\sqrt{TW_{-1}}}{Gm}.$$

Integrating $\sqrt{t}$ gives the factor $2T^{3/2}/3$ in (D.5). The same estimate holds uniformly in θ_{-i}, so averaging over these ages completes the proof. □

Combining the comparison plan and the discovery bound yields a positive learning gap. Define

$$\mathcal{R}_G = \frac{1}{2}\left\{1 - \frac{1}{m} - \frac{G}{3m}\sqrt{W_1} - \frac{\sqrt{W_{-1}}}{G^3 m}\right\}.$$

Lemmas D.1–D.3, together with $T = G^4$ and $n = KG$, imply

$$\mathbb{E}_\theta\{T\mathrm{UB}(R^\theta) - S_T\} \geq \left\{\mathcal{R}_G - \frac{1}{2}\sqrt{\frac{v}{KG}}\right\} n\sqrt{T}. \tag{D.8}$$

After subtracting the planning allowance T, the gap remains positive for sufficiently large G. Indeed, as $G \to \infty$, we have $m/G \to 5/4$, $v = O(1)$, and $W_{-1} = O(G^2)$, since j/G stays bounded away from zero and two. Moreover, uniformly over $j \in \mathcal{J}$,

$$b_j^2 = \frac{G^2}{\min\{j(2G-j), (j+1)(2G-j-1)\}} = \frac{1}{(j/G)(2-j/G)} + O(G^{-1}).$$

The Riemann sum for W_1 therefore satisfies

$$W_1 \longrightarrow \int_{3/8}^{13/8} \frac{dx}{x^2(2-x)} = \frac{1}{2}\log\frac{13}{3} + \frac{40}{39} < \frac{9}{4}.$$

Thus $\liminf_{G\to\infty} \mathcal{R}_G \geq 3/10$. The comparison loss in (D.8) tends to zero, whereas $T/(n\sqrt{T}) = 1/16$. Consequently,

$$\liminf_{G\to\infty} \frac{\mathbb{E}_\theta[T\{\mathrm{UB}(R^\theta) - 1\} - S_T]}{n\sqrt{T}} \geq \frac{3}{10} - \frac{1}{16} = \frac{19}{80} > 0.$$

An absolute $c > 0$ therefore gives the desired lower bound for all sufficiently large G. For each such G, at least one fixed hidden-age vector attains the prior-average bound. Scaling rewards and observations by L proves Theorem 4 along the growing instance family in (D.1).

E Experimental Details and Additional Results

This appendix specifies the experiments in Section 5 and examines capacity pressure, recovery scales, learning horizons, feedback noise, and replanning frequency. The accompanying numeric supplement contains individual-seed results, complete evaluation windows, and configurations.

E.1 Experimental setup

The central setting is $(n, K, D) = (2048, 128, 128)$, with $L = 1$ and initial ages one. Each family has three paired workload seeds. Offline and online methods use the same reward matrices; all methods within a workload share its initial ages and evaluation benchmark.

Reward curves. A shared draw of latent attributes generates three within-arm families. Half the arms have four stages and half have five, in random order. For an arm with m stages, draw terminal reward $A \sim \mathrm{Unif}[0.6, 1]$ and terminal recovery age $E = \mathrm{round}(\exp(U))$, where $U \sim \mathrm{Unif}[\log 16, \log D]$. Draw independent $g_j = 0.5 + \mathrm{Exp}(1)$ and $v_j \sim \mathrm{Exp}(1)$, and set

$$s_j = 3 + (E - 3m)\frac{g_j}{\sum_\ell g_\ell}, \qquad w_j = 0.05 + (1 - 0.05m)\frac{v_j}{\sum_\ell v_\ell}.$$

Thus the stage lengths sum to E and the positive reward increments sum to one. For a stage ending at $b_j = \sum_{\ell \leq j} s_\ell$, choose $z_j = \max\{3, s_j U_j\}$ with $U_j \sim \mathrm{Unif}[0.5, 0.9]$. A smooth increment uses

$$x_j(d) = \min\{1, \max\{0, (d - b_j + z_j)/z_j\}\}, \qquad S_j(x) \in \{3x^2 - 2x^3,\ 6x^2 - 8x^3 + 3x^4,\ 4x^3 - 3x^4\},$$

with the shape drawn uniformly. A discrete increment is $\mathbf{1}\{d \geq b_j - z_j/2\}$. The arm's reward is A times the weighted sum of its increments, rounded to the nearest 10^{-6}.

Continuous arms use smooth increments throughout, and Staircase arms use discrete increments. Hybrid arms have two smooth stages if $m = 4$, or two or three with equal probability if $m = 5$, at uniformly chosen positions. Mix independently shuffles 683 Continuous, 683 Staircase, and 682 Hybrid arms into one population. The curves are constant after D in these test instances. Theoretical guarantees allow general bounded, nondecreasing curves.

Instance ranges and pairing. The offline capacity scan fixes $D = 128$, takes $K \in \{64, 96, 128, 192, 256, 384, 512, 640, 768, 896, 1024\}$, and sets $n = 16K$. Larger populations are built from independently generated 2048-arm blocks, taking the first n arms; the first block is the central input. For the pressure and recovery scans, $p_i^\star$ is the longest maximizer of $R_i(p)/p$, and $\rho = (\sum_i 1/p_i^\star)/K$. The pressure scan fixes $K = D = 128$ and takes

$$\rho \in \{1, 1.25, 1.5, 1.75, 2, 2.5, 3, 4, 5, 6, 8, 10\}.$$

Within each family and seed, populations are nested prefixes of a common pool of independently generated arms. The prefix length n is chosen to match the target demand, using the nearest feasible length.

The recovery scan fixes $K = 128$ and targets $\rho = 2$, with

$$D \in \{64, 96, 128, 160, 192, 224, 256\}.$$

It evaluates the same latent curves at $128d/D$, preserving stage heights and relative locations, and adjusts n to match demand. This is a load-matched comparison across recovery scales. The maximum relative discrepancy between achieved and target pressure is 0.126%. All comparisons use four families and three paired seeds. The capacity scan has 132 environments and 396 method trajectories. The two sensitivity scans have 216 environments and 648 method trajectories, counting the shared $D = 128, \rho = 2$ conditions once.

The central online comparison uses $T = 16384$ and three seeds per family. The online load-pressure comparison uses Mix, $K = D = 128$, $T = 32768$, and $\rho \in \{1, 2, 3, 5, 10\}$. It reuses the corresponding nested populations from the offline pressure scan, with three seeds and two methods, MBI-KL and RTI-A, giving 30 trajectories. Population size varies from 1244 to 12475 arms across these workloads. The Mix horizon comparison ($T = 16384, 32768$) and MBI-KL block-multiplier comparison use two prespecified seeds; the feedback comparison uses three seeds. These three sensitivity studies fix $n = 2048$ and $K = D = 128$. Each comparison uses paired reward matrices and policy/feedback seeds. Seed values and generator offsets are supplied in the numeric supplement.

Feedback and evaluation. Online learners receive observations from their executed pulls, while the evaluator uses the true conditional means. Bernoulli observations have mean $\mu = R_i(d)$. The lower-noise control uses a symmetric triangular distribution with mode μ and support $[\mu - r, \mu + r]$, where $r = \min\{\mu, 1 - \mu\}$. Both use the same reproducible date–arm random field. Continuous observations are stored as float64 values and supplied to estimation as their exact binary fractions.

Offline central, capacity, and pressure comparisons use $T = 32768$. The load-matched recovery scan uses $T = 256D$, keeping the horizon proportional to the recovery scale. Offline results report UB minus mean conditional reward over rounds $1, \ldots, T$; Figure 5 additionally evaluates rounds $129, \ldots, T$ to examine performance after startup. Online loss also averages from round one and includes exploration, waiting, and execution. UB is the renewal relaxation benchmark; the exact hard-cap optimum is OPT. Seed means, standard deviations, and ranges describe variation on the developed workloads.

E.2 Implementation details

MBI and MBI-KL. MBI implements Appendix A with exact LP time shares, the residual root menu $q(p) = \min\{q \geq p : \gcd(q, 30) = 1\}$, support-aware rational buffers, dependent quotas, and raw-gap filtering. It returns the first candidate whose score is at least $[h[\mathrm{UB}_{100n} - 0.99]_+ - n]_+$. For known saturated curves, dominated long-period LP columns can be removed while preserving the renewal optimum. MBI-KL uses $\delta = 1/20$ and block multiplier 8 in (C.11), with the confidence curves, balanced allocation, and continuation rule of Appendix C. Planning incorporates observations after each completed block and retains separate statistics for each age. MBI-KL operates with T and D unknown. The block-multiplier study changes only this multiplier to 4 or 16.

PPP: offline planning. We implement the nine-candidate procedure in Section 7.1 of the extended preprint of Simchi-Levi et al. (2025), using arXiv:2106.14813v2 for its experimental specification. For each $a \in \{\lfloor\sqrt{K}\rfloor, \lceil\sqrt{K}\rceil\}$, construct three schedules by moving the exceptional mixed frequency to its faster endpoint, keeping its original frequency, or using its slower endpoint. An idle endpoint remains idle. Periods are rounded to the corresponding union of odd-base dyadic chains,

each chain is packed into unit-load groups, and the best K groups are retained. Divisibility gives a residue schedule within each group. These are six logical candidates, including duplicates when K is a square.

Three further candidates solve the integer problem with capacity K on the individual chains $\{(2a-1)2^{\ell} : \ell \geq 0\}$, for $a = 1, 2, 3$. Each chain ends at its first period $P \geq D$, which suffices for saturated rewards. These problems are solved by exact integer dynamic programming with weights P/d. PPP selects the candidate with the largest exact long-run reward rate.

PPP: online learning. The five-candidate procedure in the extended preprint's Section 7.2 uses native Hoeffding estimates

$$U_i(d) = \min\left\{\widehat{R}_i(d) + \sqrt{\frac{2\log(KT)}{\max\{1, N_i(d)\}}}, 1\right\}.$$

For each of the two values of a above, solve the integer problem on the union-of-chains menu with capacity $K+1$ and apply the original rounding and scheduling routine. The other three candidates solve the single-chain problems with capacity K. The feasible schedule with the largest UCB reward rate is selected. The phase length is $\phi = \max\{2, \lceil\sqrt{T/\log(K+1)}\rceil\}$, and periods are at most $\lfloor\phi/2\rfloor$. At the central setting these are 59 and 29, respectively. Counts persist across phases; estimation uses observations after each arm's first pull in a phase, at its assigned period.

Union-menu integer problems have certified absolute objective error at most 10^{-4}. For candidate capacity C and menu $\mathcal{D}$, a nonnegative price λ provides the upper bound

$$C\lambda + \sum_i \max\left\{0, \max_{d\in\mathcal{D}} \frac{U_i(d) - \lambda}{d}\right\}.$$

Rational arithmetic certifies both feasibility and the gap between this bound and the returned objective. If this screening is insufficient, integer optimization supplies the certificate. Single-chain problems use exact dynamic programming. The tolerance applies to each planning objective.

RTI and RTI-A. Known-curve RTI follows Algorithm 1 of Papadigenopoulos et al. (2022): sample an arm's period using its LP time shares and an independent uniform phase. If requests exceed capacity, retain the K largest rewards at the actual ages, breaking ties by arm identifier. Its virtual periodic clock continues after a clipped request.

For the original learning prescription used as a reference in Section 5, we instantiate the asymptotic error scale with multiplier one:

$$\varepsilon = \min\{(nD^2\log(nDT)/(KT))^{1/3}, 1 - 1/\max(2,T)\},$$
$$\delta_T = 1/\max(2,T), \qquad m = \left\lceil \frac{\log(2nD/\delta_T)}{2\varepsilon^2} \right\rceil.$$

Sequential batches of at most K arms then require $\lceil n/K\rceil\{1 + mD(D+1)/2\}$ exploration rounds. At the central setting, $m = 12$ gives 1,585,168 rounds. This is a numerical instantiation of the asymptotic prescription.

RTI-A instead collects one target observation per arm at every age $d \in \{1, \ldots, D\}$, then commits to RTI using fitted reward curves. With $g = \lceil n/K\rceil$ batches, pipelined exploration takes

$$E = gD + \sum_{d=1}^{D} d\lceil g/d\rceil$$

rounds. At each age d, waves contain at most d batches; successive batches are reset one round apart and each is pulled again after d rounds. Sharing waiting rounds preserves the target ages and capacity. All reset rewards count toward evaluation; only the target-age observations enter the fit. At commitment, count-weighted isotonic regression fits each arm's means by the pooled adjacent violators algorithm. The fitted curves drive RTI's LP, period and phase sampling, and actual-age estimated-reward top-K rule. Physical ages persist, and the fitted estimates remain fixed during execution.

RTI-A receives D and uses the same one-observation budget in every comparison. Its empirical finite-horizon adaptation has a different sampling budget from the original regret analysis. Central exploration takes $E = 10493$ rounds, so commitment begins at round 10494. A condition is feasible for this comparison when $E < T$.

Computation and reproducibility. Recorded timings reflect shared-machine runs and are supplied as computational-cost diagnostics. Two triangular-feedback MBI-KL runs required approximately 6280–6896 planner seconds, compared with 625–647 seconds for their Bernoulli counterparts. This cost motivates the replanning comparison below. The numeric supplement retains per-run timing coverage, input and trajectory identities, replay checks, and all evaluation windows. The scripts `render-unified-figures.py` and `render-offline-additions.py`, together with `render-pressure-focus.py`, generate the displayed results from the numeric exports in experiment-data/.

E.3 Additional offline results

Figure 4 separates the capacity-scaling comparison by family. MBI retains a small absolute loss in every family. Mean preferred demand/capacity stays between 1.632 and 1.646, and all 132 workloads have positive capacity prices.

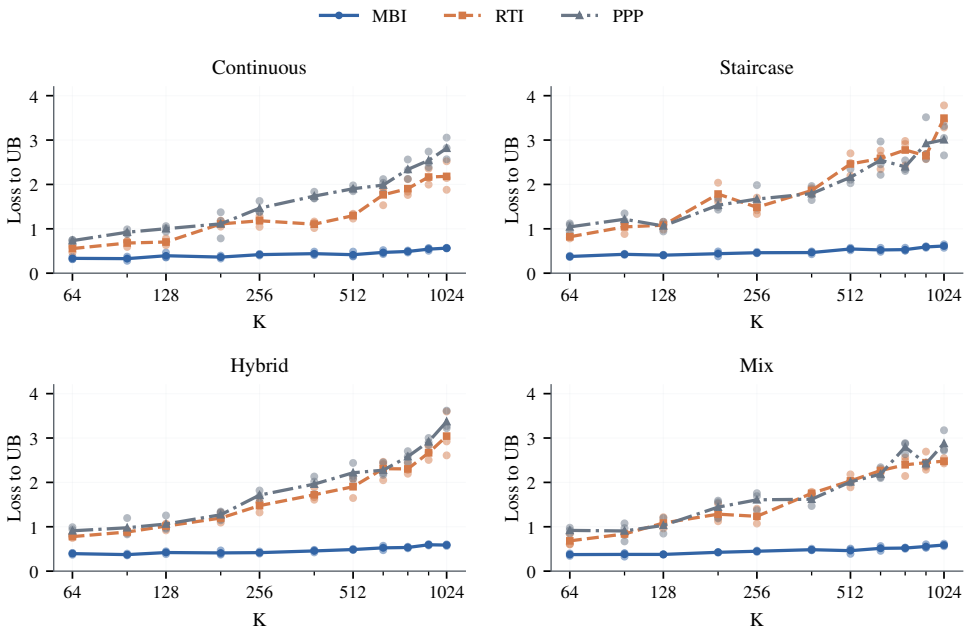


Figure 4: Full-horizon offline loss by recovery family, with $n = 16K$, $D = 128$, and $T = 32768$. Lines show three-seed means and points show individual seeds.

At the central setting, full-horizon loss averaged equally over the four families is 0.399 for MBI, 0.970 for RTI, and 1.040 for PPP. MBI has the smallest full-horizon loss on all 132 scaling workloads.

Capacity scaling after startup. Figure 5 evaluates the same capacity-scaling trajectories after excluding the first $D = 128$ rounds. The loss over the remaining rounds is

$$\mathrm{UB}(R) - \frac{1}{T-128} \sum_{t=129}^{T} \sum_{i \in A_t} R_i(d_i(t)).$$

Averaged equally over the four families, MBI's three-seed mean loss stays between 0.356 and 0.388 across the eleven capacities.

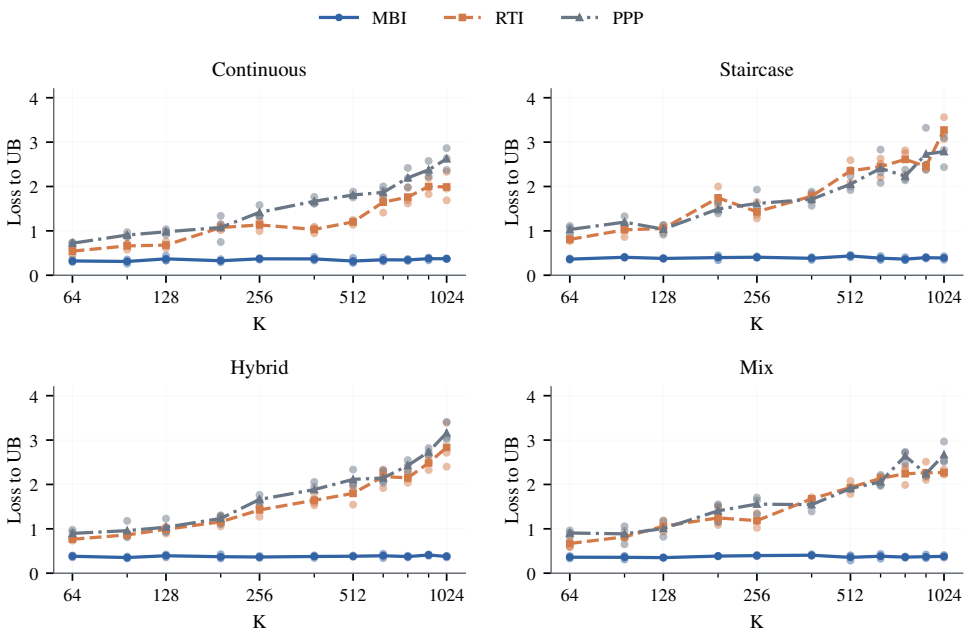


Figure 5: Offline loss by recovery family over rounds 129–32768, with $n = 16K$ and $D = 128$. The first 128 rounds are excluded from evaluation; the original trajectories and arm states are retained. Lines show three-seed means and points show individual seeds.

Longer horizon at large capacity. Table 2 compares two horizons at $K = 1024$. MBI has the lowest full-horizon loss in every family. Its average over the four families decreases from 0.588 to 0.493, while RTI and PPP attain 2.696 and 2.914 at $T = 65536$. MBI's calendar is constructed for each horizon; PPP and RTI retain identical trajectories over the first 32768 rounds.

Table 2: Full-horizon offline loss at $K = 1024$, $n = 16384$, and $D = 128$. Entries are means $\pm$ sample standard deviations over three paired seeds. For the overall rows, each seed is first averaged equally over the four families.

Family	T	MBI	RTI	PPP
Continuous	32768	0.564 ± 0.007	2.182 ± 0.323	2.815 ± 0.244
	65536	0.468 ± 0.011	2.086 ± 0.322	2.720 ± 0.245
Staircase	32768	0.612 ± 0.039	3.494 ± 0.257	3.006 ± 0.333
	65536	0.501 ± 0.039	3.385 ± 0.259	2.896 ± 0.333
Hybrid	32768	0.587 ± 0.017	3.043 ± 0.505	3.370 ± 0.215
	65536	0.526 ± 0.049	2.940 ± 0.504	3.267 ± 0.216
Mix	32768	0.588 ± 0.029	2.478 ± 0.063	2.878 ± 0.258
	65536	0.476 ± 0.022	2.375 ± 0.064	2.774 ± 0.257
Overall	32768	0.588 ± 0.009	2.800 ± 0.030	3.017 ± 0.181
	65536	0.493 ± 0.009	2.696 ± 0.030	2.914 ± 0.181

Capacity pressure. Figure 6 increases demand at fixed capacity and recovery scale. Across the 48 family–pressure settings, MBI's three-seed mean loss ranges from 0.301 to 0.608, compared with 0.349–4.009 for RTI and 0.769–2.440 for PPP. MBI remains below both baselines at every setting. Its loss increases modestly as pressure rises, while the baselines show larger overall increases. PPP also has local decreases at high pressure, yielding a nonmonotone response.

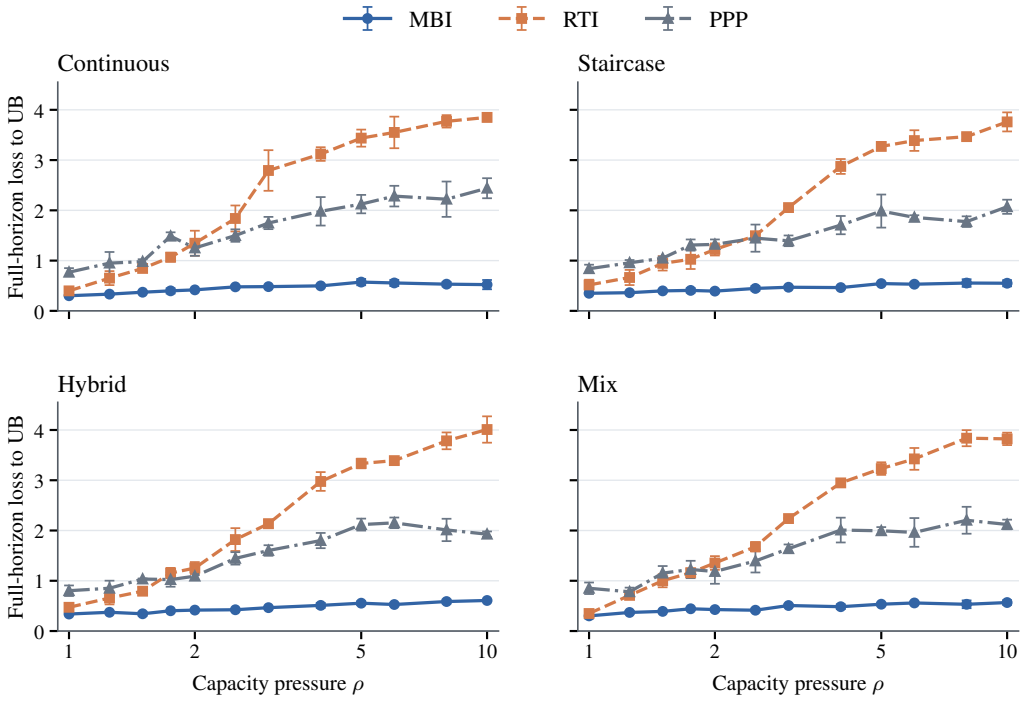


Figure 6: Full-horizon loss under capacity pressure $\rho = \text{demand}/K$, with $K = D = 128$ and $T = 32768$. The horizontal axis is logarithmic. Lines show three-seed means and error bars show sample standard deviations.

Recovery scale at matched pressure. Figure 7 varies D while maintaining $\rho \approx 2$. MBI's three-seed mean loss remains in 0.390–0.469, compared with 1.074–1.390 for RTI and 0.867–1.695 for PPP. MBI remains best at every setting and shows little sensitivity to recovery scale at matched pressure. The capacity-weighted median and 90th percentile of LP periods increase across all seven values of D in every family–seed pair, confirming that the selected waiting periods lengthen with recovery.

E.4 Additional online results

Central comparison. Figure 3 presents the full-horizon comparison and the Mix learning paths. Figure 8 shows learning paths on all four families. Table 3 compares the full horizon with the common window after RTI-A's commitment. MBI-KL has the lowest full-horizon loss on all 12 workloads, while RTI-A has the lowest loss in the shared later window. The comparison separates total learning cost from performance after exploration; the primary metric remains the average over all rounds.

Table 3: Central online loss, averaged over three paired seeds. Both windows are identical across learners and computed from the same uninterrupted trajectories.

	Full horizon			Rounds 10494–16384		
Family	MBI-KL	RTI-A	PPP	MBI-KL	RTI-A	PPP
Continuous	19.290	29.333	34.248	18.029	13.557	21.397
Staircase	25.039	34.773	40.893	23.184	15.946	27.326
Hybrid	22.689	32.702	38.136	21.174	15.574	24.890
Mix	22.533	32.511	37.954	20.995	15.338	24.699

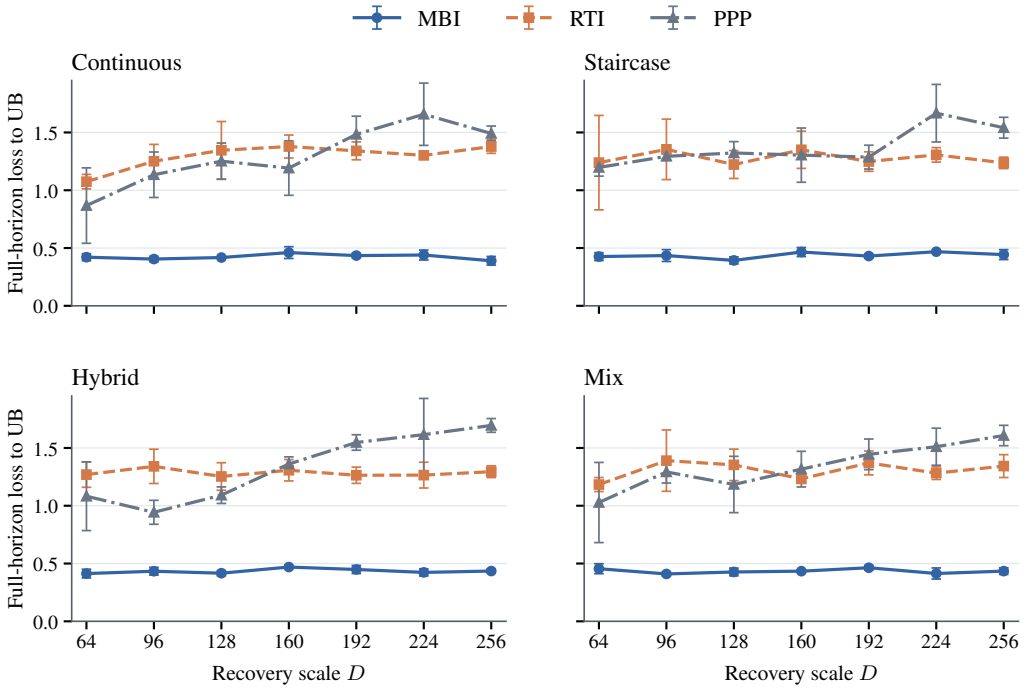


Figure 7: Full-horizon loss across recovery scales at matched pressure, with $K = 128$, $\rho \approx 2$, and $T = 256D$. Population size n is adjusted to maintain pressure. Lines show three-seed means and error bars show sample standard deviations.

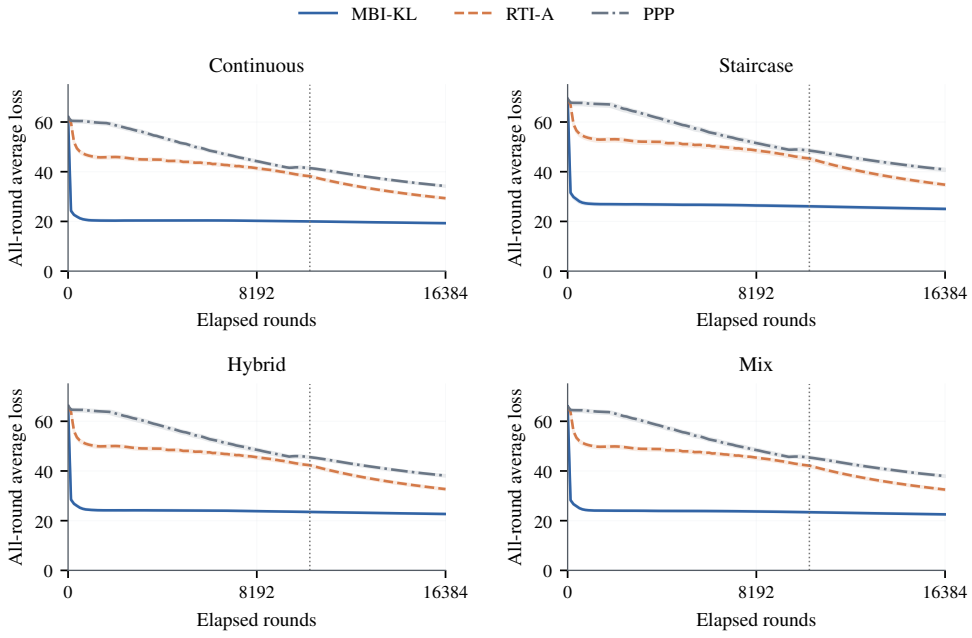


Figure 8: All-round average loss on each recovery family at $T = 16384$. Shading shows the three-seed range. The dotted line marks RTI-A's commitment at round 10494.

Load pressure. Figure 9 and Table 4 compare MBI-KL and RTI-A at the five pressure targets. MBI-KL has lower full-horizon loss on all 15 paired workloads. The three-seed mean reduction relative to RTI-A is 17.8%, 14.0%, 16.7%, 37.7%, and 56.7%, respectively. Its loss is nonmonotone in pressure, peaking near $\rho = 3$.

RTI-A explores arm–age pairs in descending age order. The exploration duration E grows with the number of batches: it uses 9605 rounds at $\rho = 1$ and 28555 rounds at $\rho = 10$. For each workload, both methods are also evaluated on the identical rounds $E + 1, \ldots, T$. RTI-A has lower loss in this common later window at the first four pressure levels, whereas MBI-KL has lower loss at $\rho = 10$, on every paired seed. At the highest pressure, this window contains 4213 rounds; the final quarter still includes RTI-A exploration. All quarter, half-horizon, and common-window results are retained in the numeric supplement.

The comparison varies population size to achieve the target pressure at fixed K, D, and T. It measures the complete learning procedures, including exploration, estimation, and scheduling.

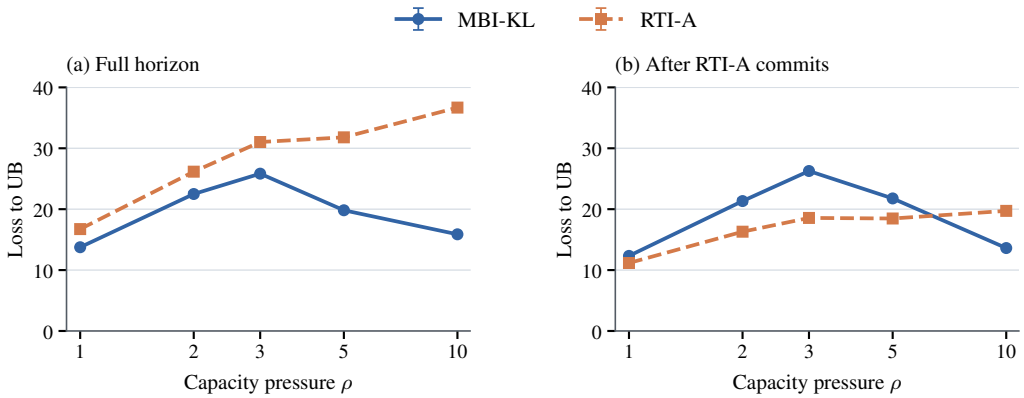


Figure 9: Online load pressure on Mix with $K = D = 128$ and $T = 32768$. (a) Full-horizon loss. (b) Loss on the common window after RTI-A's commitment, whose length depends on the workload. Lines show three-seed means and error bars show sample standard deviations; population size is adjusted to match each pressure target.

Table 4: Mix online pressure results: mean loss ± sample standard deviation over three paired seeds. The last column gives the fraction of the horizon used by RTI-A exploration.

	Full horizon		After RTI-A commits		
ρ	MBI-KL	RTI-A	MBI-KL	RTI-A	E/T
1	13.751 ± 0.036	16.731 ± 0.390	12.329 ± 0.054	11.161 ± 0.372	29.31%
2	22.490 ± 0.362	26.160 ± 0.255	21.332 ± 0.411	16.303 ± 0.391	33.91%
3	25.832 ± 0.106	31.028 ± 0.088	26.268 ± 0.089	18.562 ± 0.152	39.02%
5	19.822 ± 0.135	31.797 ± 0.229	21.765 ± 0.072	18.471 ± 0.175	50.23–50.79%
10	15.874 ± 0.070	36.681 ± 0.058	13.618 ± 0.067	19.722 ± 0.374	87.14%

Horizon. Table 5 doubles the Mix horizon with the same two seeds. All methods improve in full-horizon loss. RTI-A's exploration remains 10493 rounds, so its horizon share falls from 64.04% to 32.02%, and its loss decreases from 32.726 to 24.076. MBI-KL retains the lowest full-horizon loss, while RTI-A has the lowest final-quarter loss.

Table 5: Mix horizon sensitivity, two paired seeds. RTI-A uses one target observation per arm–age pair at both horizons.

Method	T	Full loss	Last quarter	Commit
MBI-KL	16384	22.619	20.785	–
	32768	20.966	18.504	–
RTI-A	16384	32.726	15.423	10494
	32768	24.076	15.425	10494
PPP	16384	38.161	23.642	–
	32768	30.782	22.172	–

Feedback noise. Table 6 preserves the reward means and bound while reducing observation noise. MBI-KL and PPP change little in full-horizon loss. RTI-A improves substantially, especially after exploration, revealing different sensitivity to feedback noise across these complete learning procedures.

Table 6: Mix feedback sensitivity, three paired seeds.

	Full horizon		Final quarter	
Method	Bernoulli	Triangular	Bernoulli	Triangular
MBI-KL	22.533	22.577	20.756	20.926
RTI-A	32.511	29.291	15.331	6.356
PPP	37.954	37.908	23.544	23.494

Replanning frequency. Table 7 varies only MBI-KL's block multiplier. Reducing it from 8 to 4 lowers full-horizon loss by about 0.68 and increases planning calls from 275 to 545. Increasing it to 16 reduces calls to 141 but raises loss by about 2.25. This quantifies the reward–computation trade-off around the default multiplier 8.

Table 7: MBI-KL block-multiplier sensitivity on Mix, two paired seeds.

Block multiplier	Full loss	Last quarter	Replans
4	21.938	20.423	545
8	22.619	20.785	275
16	24.867	22.898	141